Teamcenter™

Deployment Guide

Teamcenter™

Deployment Guide

11.2.1

This product is intended for use only as described in this document. Siemens PLM Software cannot be responsible for the proper functioning of features and parameters not described.

Manual History

Manual Revision	Teamcenter Engineering Version	Publication Date	Notes
A	8.1	January 2003	
B	9.1	October 2004	
C	9.1.2	June 2005	Added AIWS to Web Chapter.
D	9.1.2	July 2005	Corrected CPU requirement calculations. Updated AIWS information.
E	10.0	January 2006	Updated for V10 (Tc Eng. 2005)

Manual Revision	Teamcenter Version	Publication Date	Notes
F	Tc 2007.1	October 2007	Adapted for the Teamcenter unified platform. Updated with Tc2007.1 sizing requirements. Oracle RAC deployment information added
G	Tc 8.0	June 2009	Updated for Tc 8.0
H	Tc 8.0	September 2010	Updated with information for determining additional indexes needed for SQL Server (PR 6243629)
J	Tc 8.3	November 2010	Updated for Tc 8.3 (including Tc 8.1 revisions) DB2 server information added Proxy Server information added Integration for NX and Visualization Deployments chapter removed; now covered by NX documentation.
K	Tc 8.3	July 2011 (not published)	Updated with pom_backpointer guidelines (PR 6438112) Updated SQL Server tuning guidelines per Microsoft
M	Tc 9.1	April 2012	Updated for Tc UA 9.1 Obsolete AIWS information removed

Manual Revision	Teamcenter Version	Publication Date	Notes
N	Tc 9.1	August 2012	Minor typographical corrections and updates. Incorporated tcserver 32v64 bit comparison table from field feedback.
P	Tc 9.1	May 2013	Minor updates Added chapter on Cloud deployments Added reference to NetApp deployment and backup integration guides.
R	Tc 9.1	September 2013	Added notes explaining the use of ‘N/A’, Not Available
S	Tc 10.1	October 2013	Updated for Tc UA 10.1
T	Tc 10.1	November 2013	Minor typographical corrections and updates. Added *SQL Server CPU, Memory, and Swap Usage* table to section 5.3.6.2. Corrected database CPU data
U	Tc 10.1	January 2014	Minor typographical corrections and updates.
V	Tc 10.1	April 2014	Update suggested tables for keep pool Removed Oracle RAC install / configuration guidelines.
W	Tc 10.1	June 2014	Update Multi-site chapter for 10.1 Add Oracle Commit_No_Wait guidelines Removed specific Oracle version from configuration guidelines
X	Tc 10.1	March 2015	Corrected Table 5-2 Corrected reference to 4-node RAC support
Y	Tc 11.2.1	January 2016	Updated for Tc UA 11.2.1
Z	Tc 11.2.1	September 2016	Updated Oracle optimizer update recommendations per TC POM team. Added notice of discontinued DB2 support. Added placeholder for Active Workspace. Minor updates.
AA	Tc 11.2.1	February 2017	Added Rich Client server sizing information. Updated Oracle RAC information.

Manual Revision	Teamcenter Version	Publication Date	Notes
AB	11.2.1	February 2018	Removed Function-Based Indexes guidance per HK Minor edits Added Azure information to Cloud chapter for TC 11.2.3 and AW 3.3. Sections 9.4.1and 9.4.2 provided by ST & GB
AC	11.2.1	April 2018	Updated the URL link to Network Performance Tuning Added Microsoft Azure information to the Cloud Deployments Chapter
AD	11.2.1	June 2018	Corrected CPU sizing information in the Enterprise Server Chapter.

Contents

Figures

Tables

Preface

This guide provides general guidelines and best practices for a Teamcenter™ system administrator to follow when deploying a new, or upgrading an existing Teamcenter system.

This guide supplements the installation and configuration documentation, which resides on the Documentation CD-ROM and *Installation/Configuration* sections of the online help in the *Teamcenter Help Library*. It does not replace them.

NOTE

If you find information in this Deployment Guide to be inconsistent with that found in the *Teamcenter Help Library* or the PDF installation instructions on the Teamcenter CD-ROM, contact the Global Technical Access Center (GTAC) for clarification.

Audience

This guide is for experienced Teamcenter system administrators who have performed Teamcenter installation and configuration tasks and are familiar with the concepts taught in the following Teamcenter Engineering or Teamcenter training courses:

- ❖ Introduction to Teamcenter (TR25100)
- ❖ Using Teamcenter (TR25150)
- ❖ Installation (TR25350)
- ❖ Application and Data Model Administration (TR25460)
- ❖ Customization (TR25540, TC 11.2)

Organization

This guide is organized as follows:

Chapter 1	***Managing Teamcenter Upgrades*** outlines version-specific installation and upgrade issues important to the system administrator and a review of best practices for migrating between major versions of Teamcenter. This chapter has not been updated for Tc 11.2.1.
Chapter 2	***Managing Client Deployments*** discusses deployment of the Teamcenter Rich (Java) and Thin (Browser) Clients.
Chapter 3	***Managing Web Server Deployments*** includes information to size and tune a representative Web Application server product supported by Teamcenter. Information about using Web Application servers with firewalls and Application Interface Web Service deployments is included.
Chapter 4	***Managing Enterprise Server Deployments*** discusses deployment of Teamcenter server pools (business logic tier).
Chapter 5	***Managing Database Server Deployments*** covers Oracle, Microsoft SQL Server, and DB2 database configuration, tuning, and maintenance. Deployment of Teamcenter in an installed Oracle RAC environment is also presented.
Chapter 6	***Managing FMS & Volume Server Deployments*** addresses management issues of Teamcenter volume servers, the File Management System (FMS), and heterogeneous implementations.
Chapter 7	***Managing Classic Multi-site Deployments*** provides performance, sizing, and throughput information for Teamcenter Multi-site deployments.
Chapter 8	Purposely Omitted
Chapter 9	***Managing Teamcenter Cloud Deployments*** provides information related to Teamcenter deployments in supported cloud environments.
Chapter 10	***Improving Teamcenter Performance*** provides information related to general system-wide performance issues not covered in other chapters. This chapter has not been updated for Tc 11.2.1.

= updated for this release.

= new for this release

Glossary	The ***Glossary*** defines Teamcenter terminology.
Appendix A	***Sample User Scenarios*** to help plan and define system usage profiles.
Appendix B	***Sample Usage Profiles*** to help plan and define system usage profiles.
Appendix C	***Teamcenter Operational Maintenance*** lists several system operational maintenance utilities along with recommended frequency and potential impact to users and data.

Conventions

This guide uses the conventions described in the following sections:

1.1.1 *Names and Values*

This manual represents system names, file names, and values in fonts that help you interpret the name or value. For example:

The file name is **pom_schema_***server-name_sid*.

The conventions for names and values are:

Bold	Bold font represents unvarying text or numbers within a name or value. Capitalization is as it appears. In the preceding example, **pom_schema_** identifies an unvarying portion of the name.
Italic	Italic font represents text or numbers that vary. The characters in italic text describe the entry. Letters are shown in lowercase, but the varying text may include uppercase letters. In the preceding example, *server-name* and *sid* represent varying portions of the name.
text-text	A hyphen separates two words that describe a single entry. In the preceding example, *server-name* is a single value.

For example, the name of the **pom_schema_***server-name_sid* file might be:

pom_schema_Oraprod1_tcua

1.1.2 *Command Line Entries, File Contents, and Code*

This manual represents command line input and output, the contents of system files, and computer code in fonts that help you understand how to enter text or to interpret displayed text. For example, the following line represents a command entry:

```
-regen_schema_file user-name password group
```

The conventions for command line entries, file contents, and code are:

`Monospace`	Monospace font represents text or numbers you enter on a command line, the computer's response, the contents of system files, and computer code.

	Capitalization and spacing are shown exactly as you must enter the characters or as the computer displays the characters. In the preceding example, `-regen_schema_file` identifies an unvarying portion of the command.
`Italic`	Italic font represents text or numbers that vary. The characters in italic text describe the entry. Letters are shown in lowercase, but the required text may include uppercase letters. When entering text, use the case required by the system. In the preceding example, *`user-name`*, *`password`*, and *`group`* identify varying portions of the command.
`text-text`	A hyphen separates two words that describe a single entry. In the preceding example, *`user-name`* is a single entry in the command.

The following example is a correct entry for the preceding regen_schema_file command:

`-regen_schema_file infodba` *`password`* `dba`

1.1.3 *Syntax Definitions*

This manual uses a set of conventions to define the syntax of Teamcenter commands, functions, and properties. Following is a sample syntax format:

`harvester_jt.pl` [*`bookmark-file-name bookmark-file-name ...`*]
[*`directory-name directory-name ...`*]

The conventions for syntax definitions are:

`Bold`	Bold text represents words and symbols you must enter exactly as shown. In the preceding example, you enter **`harvester_jt.pl`** exactly as shown.
`Italic`	Italic text represents arguments that you supply. In the preceding example, you supply arguments for *bookmark-file-name* and *directory-name*.
`text-text`	A hyphen separates two words that describe a single entry. In the preceding example, *bookmark-file-name* is a single argument in the command.
`[]`	Brackets represent optional values.
`...`	An ellipsis indicates that you can repeat the preceding argument.

Following are *examples* of correct syntax for the **harvester_jt.pl** command:

```
harvester_jt.pl

harvester_jt.pl assembly123.bkm

harvester_jt.pl assembly123.bkm assembly124.bkm
assembly125.bkm

harvester_jt.pl AssemblyBookmarks
```

Teamcenter Documentation Set

Siemens PLM Software provides two sources of online help for Teamcenter:

- Doc Center
  A web site that provides online help for Teamcenter and other Siemens PLM Software products. A WebKey account is required to access this site. Teamcenter online help is available on Doc Center at:

  https://www.plm.automation.siemens.com/locale/docs/tc/

- Siemens PLM Documentation Server
  A web application you install on your local network that hosts online help for Teamcenter and other Siemens PLM Software products. For information about installing Teamcenter help and the PLM Documentation Server, see *Installing Teamcenter Documentation* (**help_install_instructions.pdf**) in the Teamcenter documentation distribution media.

  Teamcenter help is available in HTML and PDF formats. You can access Teamcenter help directly using the URL to Doc Center above or a local PLM Documentation Server if you have installed it. You can also configure Teamcenter clients to open your preferred documentation source through the client Help menus.

  NOTE

  PDF help is not accessible through client Help menus.

Submitting Comments

To report documentation errors or to suggest improvements to this manual please submit an incident report to the *Global Technical Access Center* (GTAC). Use the GTAC online support tools at the following URL:

http://support.ugs.com

Software Copyright and Trademark Notices

Chapter

1 Managing Teamcenter Upgrades

Go to main Table of Contents

Chapter

1 *Managing Teamcenter Upgrades*

This chapter provides guidelines for upgrading from previous releases of Teamcenter. It also provides upgrade considerations specific to this version, if any. This chapter has not been updated from the previous Teamcenter release.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

1.1 Planning the Upgrade

If you are upgrading from an existing version of Teamcenter or Teamcenter Engineering, you should plan to spend adequate time to lay out an upgrade plan. Upgrading a critical system such as Teamcenter should not be undertaken without careful preparation.

1.1.1 *Upgrade Plan Overview*

System administrators should become thoroughly familiar with the new features of each Teamcenter release prior to deployment or upgrade. This helps you plan and eliminate surprises when the work is in progress. Also, there are sometimes limitations that apply to upgrades such as the need to upgrade to an intermediate release before upgrading to the current one.

Questions about new features or information in the *Teamcenter Release Bulletin* can be directed to the *Global Technical Access Center* (GTAC).

1.1.2 *Create a Critical Path Activity Plan*

A careful upgrade approach includes a critical path activity plan with task dependencies, anticipated durations, and required resources. Although the upgrade effort may vary according to the complexity and scope of your Teamcenter implementation, your plan should minimally address the following topics:

- Review new features in the *Teamcenter Help Library* and the *Teamcenter Release Bulletin*.
- Create precautionary backups of:
    - Executables and shared libraries (*TC_ROOT*)
    - Data directory (*TC_DATA*) and configuration files
    - Volume directories. You can use the `backup_xmlinfo` utility to get information about Teamcenter volume locations for the configured database.
    - Oracle, Microsoft SQL Server, or DB2 binaries and data/system tables
    - Customizations, utilities, scripts, integrations with other systems, and specialized, site-specific applications.
- Allow sufficient time to analyze, design, implement, test, and validate these upgrades.
- For customizations, the *Teamcenter Release Bulletin* contains information regarding any changed or obsolete ITK functions or other APIs.
- Upgrade all required applications and recommended operating system versions / patches before upgrading Teamcenter (install patches and perform kernel parameter changes as noted in *Teamcenter Release Bulletin*).
- Perform required or recommended upgrades to third-party software before upgrading Teamcenter (for example, WebLogic or Oracle). Check for Siemens Software Field Bulletins (SFBs) where applicable.
- Perform required changes in third-party application defaults or default behavior (for example, changes in MIME types).
- Consider new services/daemons. If new Teamcenter features require processes to run as a service or a daemon, decide where (on what machine(s)) they will run.
- Test upgrade procedures and execution, including test data. Validate test results and obtain sign-offs from respective departments and groups.
- If an NX upgrade is also being performed, decide whether or not to run the NX re-file utility (ugmanager_refile) for sites managing NX data.
- Prepare and review new/revised operating procedures and usage policies.
- Prepare for appropriate training and document delivery for users and system operators. This information may include new feature summaries targeted for specific groups of users.

- Create risk management and contingency plans to roll back to the previous release if the upgrade is not completed successfully or in the allotted time, or if it is determined after the system is released that something did not upgrade correctly. A suitable risk management plan describes the most likely failure scenarios, the actions to take to avoid their occurrence, and actions to take if they do occur. Rank the failure scenarios by their probability and impact, and focus on the most likely, highest impact risks.
- Change business practices that may be appropriate based on new features.

When you complete the upgrade plan, review it with the key people affected by the upgrade, such as management, user representatives, and developers. Use the plan to track the execution of the upgrade, and identify schedule conflicts (that is, tasks behind schedule) before they occur. If the number of users is significant or database size is large, it is highly recommended to implement a formal test process with user signoff.

1.1.3 *Implement an Upgrade Test System*

Many sites that customize their Teamcenter implementations maintain separate production, test, and development systems. This allows developers to promote a new customization into the test system and then roll it into production in a controlled manner, to minimize risk. But testing the upgrade on separate servers is just as important even if there are no customizations.

If possible, install the new Teamcenter release, as well as Oracle and any customizations, on a separate test system that includes a representative sample of production data and files. On that system, you can test all aspects of the upgrade without risk or impact to production users. Develop detailed UAT (User Acceptance Testing) documentation to test, validate, and obtain sign-offs before production database is upgraded. Document lessons learned during this test/validation period for resolution of issues and to validate workarounds when specific solutions are not immediately available. Often users can continue after the upgrade using temporary solutions (i.e. 'work-arounds') until issues are resolved. Then you can work with Siemens GTAC via IR/PR process for a formal resolution of issues post upgrade. Remember the user community must be adequately educated and trained with the work-arounds until formal maintenance packs are released by Siemens PLM Software.

When you have successfully completed your test of the entire test system (including Teamcenter, Oracle, data, and customizations) you can confidently upgrade the production systems.

1.1.4 *Informing Users When the System Goes Offline*

Use a carefully thought-out, critical path plan to predict when will you shut down the system and estimate how long it will be out of service. Plan to give users and management sufficient notice when and for how long the system will be unavailable.

Remind them periodically so they do not schedule critical production work when the system is expected to be offline. If there are projects that must continue while the upgrade is in progress, work with users to ensure they have exported the data they need while the system is offline, and define a plan to import their work after the upgrade is complete. Be certain users keep accurate records for any data created/modified during the system upgrade. Also, ensure there is a proven and tested plan/strategy in-place to merge this data with the rest of the production data when the upgrade is complete. Another important task in your planning is to remember to notify any global sites that may require data from the site being upgraded.

Despite your attempts to make sure all users are aware the system is shutting down, there may be some who do not log off, or who try to log in to the system while you are trying to shut it down. If so, you may need to lock users out. The procedure varies depending on your implementation, but methods include:

1. Disabling logins to the Oracle system.
2. Removing access to Teamcenter executables and Teamcenter vaults.
3. Establishing startup scripts that check for system availability through a status file that you create/update.

1.1.5 *Upgrade Plan Review*

When you complete the migration plan, review it with the key people affected by the upgrade, such as management, user representatives, and developers. Use the plan to track the execution of the upgrade, and identify schedule conflicts (that is, tasks behind schedule) before they occur.

1.2 Upgrade Considerations for this Version

1.2.1 *Upgrading Deployments with Oracle*

After every upgrade or patch to an environment using Oracle as the database, be sure to update Oracle’s optimizer statistics. This ensures that the optimizer chooses the best execution plan within a Teamcenter environment. Outdated or missing statistics can result in poor query performance.

Always analyze indexes whenever tables are analyzed. Always use the **dbms_stats.gather_schema_stats** procedure to update both the table and index statistics at the same time

Oracle’s DBMS_STATS.AUTO_SAMPLE_SIZE estimation parameter-value collects statistics faster and with nearly the accuracy of sampling 100% of the data.

You can use the DBMS_STATS.AUTO_SAMPLE_SIZE parameter-value to collect statistics provided there are no execution plan issues that result from its use. Databases with slow query executions (identified as slow query notes and/or execution plans in syslogs) should be analyzed at 100% to provide the best possible execution plan.

There are four basic commands that can be used to collect statistics.

1) Collecting statistics for an entire schema <u>where there is no expectation of problem query plans</u>:

```
exec dbms_stats.gather_schema_stats(ownname=>'infodba',
estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt=>'FOR ALL
COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true,
no_invalidate=>FALSE );
```

2) <u>Collecting the best possible statistics</u> for an entire schema:

```
exec dbms_stats.gather_schema_stats(ownname=>'infodba',
estimate_percent=>100, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true, no_invalidate=>FALSE );
```

NOTE:

You should reproduce performance issues using statistics collected at 100 percent before filing PRs associated with poor performance.

If you would like to collect statistics for a single table one of the following two commands can be used.

3) Collecting statistics on a single table <u>where there is no expectation of problem query plans</u>:

```
exec dbms_stats.gather_table_stats(ownname=>'infodba', tabname=>'<TABLE>',
estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt=>'FOR ALL
COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true,
no_invalidate=>FALSE );
```

4) <u>Collecting the best possible statistics</u> for a single table:

```
exec dbms_stats.gather_table_stats(ownname=>'infodba', tabname=>'<TABLE>',
estimate_percent=>100, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true, no_invalidate=>FALSE );
```

Chapter

2 *Managing Client Deployments*

Go to main Table of Contents

Chapter

2 Managing Client Deployments

This chapter introduces the Teamcenter Rich and Thin Clients as well as sizing and tuning tips.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- Client installation for: *Windows*, *Linux*, *Macintosh*
- Server installation for: *Windows*, *UNIX/Linux*
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

Three types of clients are supported with Teamcenter:

- A rich, Java-based client. The Rich Client provides access to all Teamcenter features and functions, requiring a Java Runtime Environment and a local (or mapped) install on the user's desktop machine. The Rich Client can either be deployed in 2-tier or 4-tier architecture.
- A thin, browser-based client. The Thin Client provides access to the most common features and functions needed by users who reference or consume Teamcenter information, and requires only a commercial browser to be installed on the machine.
- Beginning with Teamcenter 9.1.2.3, a lightweight Active Workspace client was supported with limited functionality. New features are being added to the Active Workspace client with each release

NOTE

This chapter covers rich and thin client deployments only. Active Workspace clients will be covered in a future revision.

If you are upgrading from a 32 bit version of Teamcenter note there is no 32 bit version available for TC 11.2.1. For the 64 bit TC 11.2.1 keep in mind there are a number of Teamcenter, Windows, and/or third party software parameters that may need to be adjusted for deployment or performance reasons.

The scope of this chapter will focus on the Teamcenter Rich Client since the Thin Client makes use of standard Internet web browsers.

2.1 Teamcenter Rich Client Architecture Design

The Teamcenter Rich Client can be deployed within 2-tier or 4-tier architecture on each user workstation using Teamcenter Environment Manager (TEM). Refer to the Client Installation Guides under *Installing* in the *Teamcenter Help Library*. The following two sections outline the differences.

2.1.1 *Rich Client 2-Tier Architecture*

A 2-tier Rich Client can be used in an environment where users have robust desktop workstation machines connected directly to the database and FMS servers via a LAN connection.

In the 2-tier deployment the Java based Rich Client, the Business Logic (Enterprise) tier, and supporting software execute on the users' workstations. Database, volumes, and other resource tier components execute on separate systems. Figure 2-1 below shows the components of the 2-tier Rich Client.

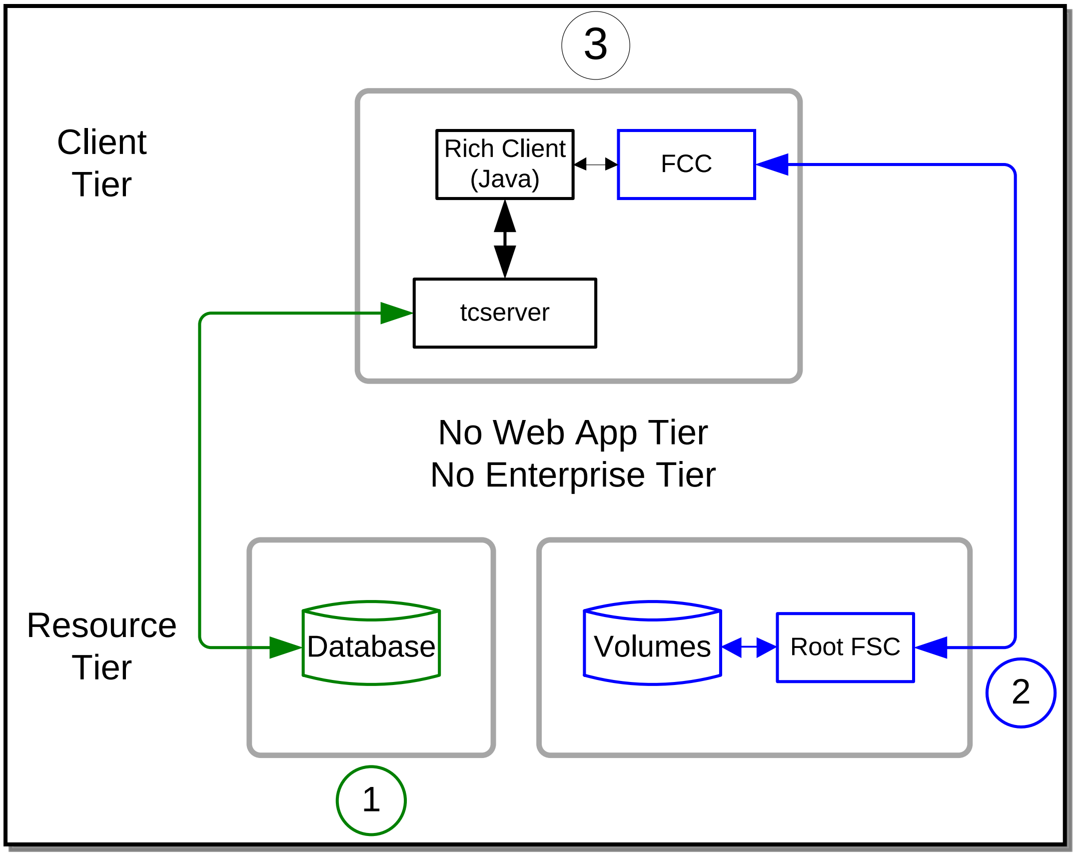


Figure 2-1, 2-Tier Rich Client Components[1]

Where:

1) Database Server (Oracle, SQL Server, or DB2)
2) Teamcenter File Management System
3) Teamcenter 2-tier Rich Client / Server (tcserver)

For more information about the Rich Client 2-tier architecture, refer to the Teamcenter Help Collection – Site Planning section of the *Installation on UNIX and Linux Servers or Installation on Windows Servers* guides.

NOTE

Operating 2-tier Rich Clients in a WAN environment is not recommended by Siemens PLM Software. The Rich Client user interface is highly interactive requiring very low latency (ping time) from the Teamcenter process (tcserver) to the database server. Latencies exceeding 5ms will likely provide unacceptable end-user response times in 2-tier deployments.

[1] Line legend: **Client-Server transfers, Database transfers, File transfers**

2.1.2 ***Rich Client 4-Tier Architecture***

In the 4-tier deployment, only the Java based Rich Client and a lightweight FCC[2] process execute on the users' workstations. The Web Application, Enterprise (or Business Logic), and Resource (Database / Volumes) tiers execute on separate systems. Figure 2-2 below shows the components of the 4-tier Rich Client. Note that multiple Web Application server instances can be deployed on one or more physical servers to facilitate a redundant, high availability Web server environment. Similarly, one or more Enterprise tier server pools (a Server Pool Manager and associated tcserver processes) can be deployed on a single or multiple machines.

[2] File Client Cache – part of the Teamcenter File Management System (FMS)

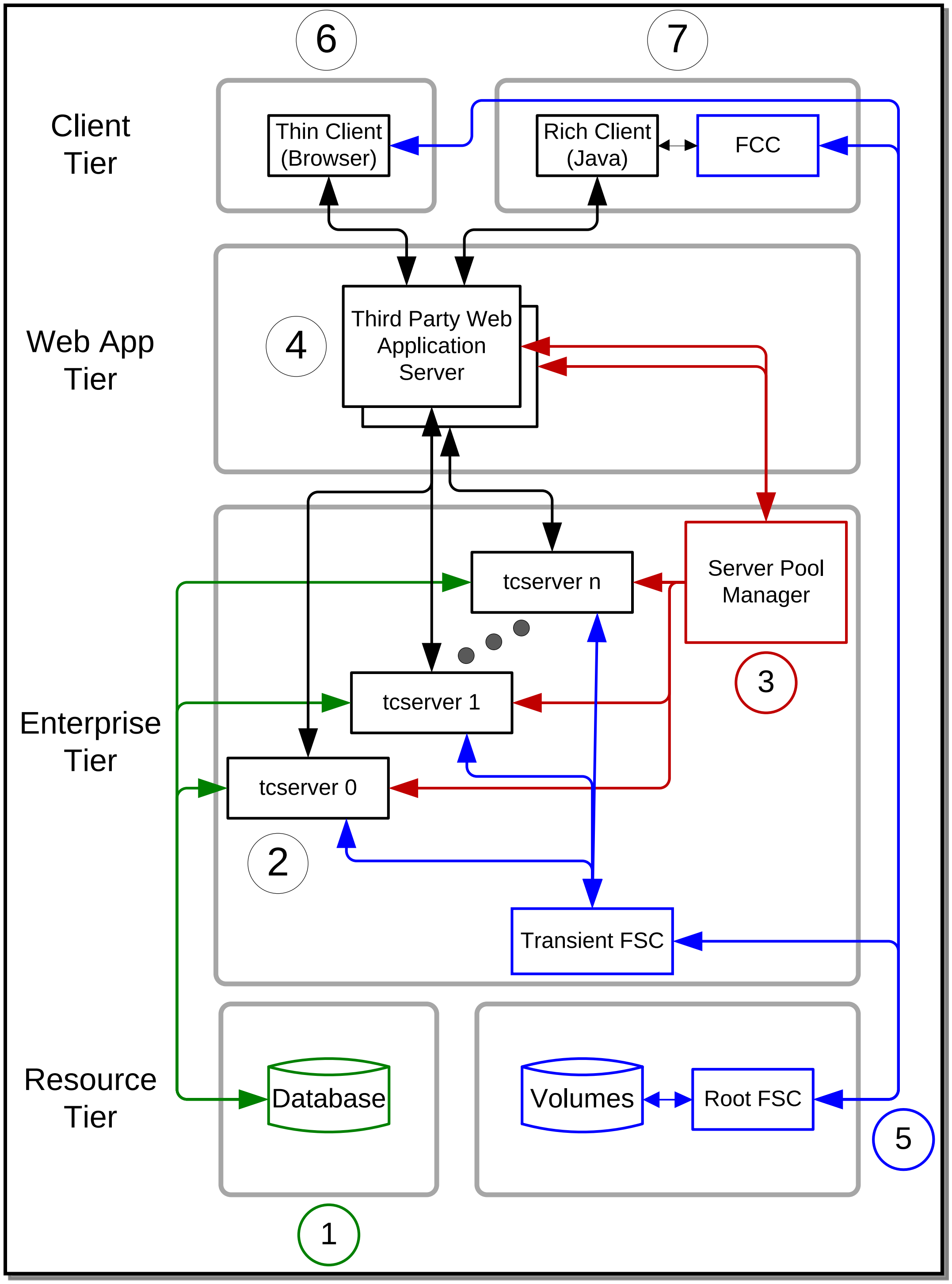


Figure 2-2, 4-Tier Rich Client Components[3]

[3] Line legend: **Client-Server transfers, Database transfers, File transfers, Session Management messaging**

Where:

1) Database Server (Oracle, SQL Server, or DB2)
2) Teamcenter Server Pool processes (tcservers)
3) Teamcenter Server Pool Manager
4) Web Application Server (e.g. WebLogic, WebSphere, JBoss, etc.)
5) Teamcenter File Management System (i.e. the root File Server Cache)
6) Teamcenter Thin (browser) Client
7) Teamcenter 4-tier Rich Client

For more information about the Rich Client 4-tier architecture, refer to the *Teamcenter Help Library – Site Planning* section of the *Installation on UNIX and Linux Servers* or *Installation on Windows Servers* guides.

See also chapters 3, *Managing Web Server Deployments*, and 4, *Managing Enterprise Server Deployments* for additional information about deploying Teamcenter 2-tier and 4-tier options.

2.2 Teamcenter Client Performance

Both the Rich and Thin Teamcenter clients make extensive use of Java technology on multiple platforms. Ensure that the JVM heap settings provide adequate memory for optimal performance. Allocate as large a heap as the machine and CPUs can handle (using '-Xms -Xmx') and still peacefully co-exist with other applications running on the client machine.

If you suspect that the JVM heap settings are contributing to poor client performance, use diagnostic tools to examine JVM events such as garbage collection (GC). Also, be sure to read and follow the GC tuning documentation for the JVM being used. Refer to Chapter 3 Managing Web Server Deployments - JVM Tuning for more detailed information.

NOTE

The default JVM runtime options may change from JVM release to release (e.g. Java 1.5 vs. Java 1.7) or OS to OS (e.g. Java stack size on Solaris vs. Windows).

2.2.1 *Rich Client Performance*

Experience at a number of customer deployments of Teamcenter have found a number of measures that may improve startup (Rich Client launch) and login performance. SFB-Teamcenter-5315, *Teamcenter Rich Client Login Performance*, available on the GTAC Solution Center, describes these measures. See also the *Configuring the rich client for startup performance* chapter in the *System Administration Guide*.

2.3 Rich Client Sizing Guidelines

NOTE

Both the environments and usage profiles used to simulate Teamcenter operations, including scenarios, transactions, number and types of files, operating systems and databases have changed since the previous Teamcenter Deployment Guide was published. ***For these reasons it is not valid to compare estimates in this revision of the guide with previous revisions.*** Use the simulation information in this chapter and extrapolate from that to estimate the target pilot environment.

Client sizing is highly dependent on usage. For example, occasional users that log into the system only to view a few parts or part properties and log out require less memory and CPU resources. Users who load large assemblies into the Structure Manager (SM) application require large amounts of memory, potentially 200-400 MB or more. Users who take advantage of multiple Teamcenter Rich Client applications (such as SM, My Worklist, Manufacturing Process Planner (MPP), Schedule Manager, and so forth) require additional memory for each application loaded.

If you are upgrading from a 32 bit version of Teamcenter note there is no 32 bit version available for TC 11.2.1. Deploying 64 bit TC 11.2.1 will allow larger sets of information to be loaded, e.g. very large BOM views, if there is adequate RAM in the system. However, 64 bit will consume 30-50% more memory for the same use cases, and possibly more CPU depending on the use case. If the system has insufficient RAM to support 64 bit Rich Client, some use cases may cause the system to start paging memory to disk, which may have a severe negative impact to performance.

NOTE

Minimum disk space and software requirements supporting Teamcenter Rich Client software can be found in the *Teamcenter Release Bulletin* (ReleaseBulletin.pdf) located in the /pdf directory of the distribution documentation CD.

2.3.1 *Sizing Client Hardware for Rich Client*

Rich Client benchmarks were performed and resource utilization of the client workstation was measured for both 2-tier and 4-tier environments, with 64 bit Teamcenter executables. The benchmarks execute a pre-defined sequence with minimal time between each step.

NOTE

The number of user types and the number of transactions measured has been increased somewhat from the previous Deployment Guide. For this reason sizing estimates documented here are not directly comparable to the previous Deployment Guide. For comparison to Teamcenter 10.1, the same activities were performance in both 10.1 and 11.2.1; the resulting workstation utilization information was recorded, and is shown in the charts below.

The workstation used for these benchmarks was a Dell Precision T1700 configured as follows:

- Intel Xeon E3-1270 v3, 3.5 GHz Quad-Core CPU
- 16 GB of physical RAM
- Microsoft 64 bit Windows 7 Enterprise with Service Pack 1

The activities in Table 2-1 were performed during each benchmark.

Table 2-1, Rich Client Benchmark Activities

User	Activity
User #1 (ps0)	1. Launch Rich Client 2. Login 3. Open Folder:Home 4. Launch Rich Client (again) 5. Login (again) 6. Open Folder:Home (again) 7. Launch Rich Client (again) 8. Login Home 9. Expand Folder:5000~items 10. Delete Item:100~items 11. Launch Application:Referencers 12. Where Referenced All Levels:55558439 001 PART 13. Where Used All Levels:13252118_002_CLIP 14. MPP Send to MPP:Mfg CC 15. MPP Accountability Check:10kx10k 16. MPP Clear Display:10kx10k 17. MPP Assign:EBOM 18. MPP Remove Assignment:EBOM 19. Logout

Table 2-1, Rich Client Benchmark Activities (continued)

User	Activity
User #2 (ug0)	1. Launch Rich Client 2. Login 3. Expand Folder:87~Items 4. Create Folder 5. Display Search Template:Item 6. Search for Item:nx6_3* 7. Load all remaining items:507~Items 8. Copy:507~items 9. Paste:507~items 10. Cut:507~items 11. Delete Folder 12. Send to PSE:nx*_3083c_6l 13. Expand BOM Below:nx*_3083c_6l 14. Display Dialog PSE Viewer:nx*_3083c_6l 15. Display in Viewer:nx*_3083c_6l 16. Send to PSE:nx*_3083c_6l~(again) 17. Expand BOM Below:nx*_3083c_6l~(again) 18. Display Dialog PSE Viewer:nx*_3083c_6l~(again) 19. Display in Viewer:nx*_3083c_6l~(again) 20. Expand BOM Below: ATO_S 21. Display in Viewer: ATO_S 22. Expand BOM Below: ATO_M 23. Display in Viewer: ATO_M 24. Expand BOM Below: ATO_L 25. Display in Viewer: ATO_L 26. Expand BOM Below: ATO_X 27. Display in Viewer: ATO_X 28. Search SM_cc:Insignia+Occ_Notes+Usage~ArchitectureID 29. Search SM_cc:Insignia+55561717_002_body_asm_srg+prox 30. Display in Viewer:405~Items~from~Insignia~Search_SM_cc 31. Send to Design Context:Insignia 32. Search DC_cc:Insignia+Occ_Notes+Usage ArchitectureID 33. Search DC_cc:Insignia+55561717_002_body_asm_srg+prox 34. Display Summary Page:Eclipse 35. 4GD Search 4GD Attribute 36. 4GD Search 4GD BoundingBox 37. 4GD Send to 4GD Insignia 38. Logout

Table 2-1, Rich Client Benchmark Activities (continued)

User	Activity
User #3 (nv0)	1. Launch Rich Client 2. Login 3. Display Dialog Create Item 4. Create Item 5. Display Dialog Create Item:(again) 6. Create Item:(again) 7. Display Dialog Create Form 8. Create Form 9. Display Dialog Create Form:(again) 10. Create Form:(again) 11. Display Dialog Create Dataset 12. Create Dataset 13. Display Dialog Create Dataset:(again) 14. Create Dataset:(again) 15. Save As New Revision 16. Save As New Revision:(again) 17. Save As New Item 18. Save As New Item:(again) 19. Delete ItemRev 20. Delete ItemRev:(again) 21. Delete Item 22. Delete Item:(again) 23. Delete Form 24. Delete Form:(again) 25. Create Requirement 26. Display Requirement Dataset 27. Edit and Exit MSWord 28. Display Requirements Manager 29. Import Specification:spec_32kb_10pages.docx 30. Export Specification:spec_32kb_10pages.docx 31. Export Requirements:spec_32kb_10pages.doc 32. Import Specification:spec_572kb_115pages.docx 33. Export Specification:spec_572kb_115pages.docx 34. Export Requirements:spec_572kb_115pages.doc 35. Logout
User #4 (wf0)	1. Launch Rich Client 2. Login 3. Expand Folder:20~items 4. Open Inbox 5. Expand Inbox Folder:Task~to~Perform 6. Expand Inbox Folder:Task~To~Track 7. Initiate Process 8. Open Inbox:(again) 9. Delete Job 10. Logout

Table 2-1, Rich Client Benchmark Activities (continued)

User	Activity
User #5 (me0)	1. Launch Rich Client 2. Login 3. MPP Manufacturing Search:E2Item_Id1_40_94_40 4. MPP Assign:BOP 5. MPP Remove Assignment:BOP 6. MPP Accountability Check:100kx5k 7. MPP Clear Display:100kx5k 8. MPP Display:BOP 9. MPP Find Product Views:nx5_ms20664c-4 10. MPP Apply Product View 11. MPP Create PV Use Current 12. MPP Apply PV Use Current 13. MPP Create PV Update Current From PV 14. MPP Apply PV Update Current From PV 15. Logout 16. Login (again) 17. MPP Send to Manufacturing EBOP 18. MPP Clone PBOP 19. MPP Recursive Resolve 20. MPP Search Operations for Allocation 21. MPP Allocate Operations to PBOP 22. MPP Open Line Balancing Chart 23. MPP Remove Allocated Operations 24. Logout (again) 25. Login (again) 26. MPP Expand Below:116690-Proc_Root 27. MPP Unpack:116690-Proc_Root 28. Logout
User #6 (an0)	1. Launch Rich Client 2. Login 3. Create Design Doc 4. CM Create Change: InContext 5. Create Dataset: CM Text 6. Check-out: Dataset 7. Check-in: Dataset 8. CM Derive Change: ECR 9. Assign Participants: ECR 10. Copy: Document Revision 11. Paste: Document Revision 12. Copy: Item Revision 13. Paste: Item Revision 14. CM Derive Change: ECN 15. Assign Participants: ECN 16. WKF Create Workflow: Approved 17. WKF Create Workflow: CM Release 18. CM Create Effectivity 19. Logout

Key system utilization metrics were collected during the execution of the benchmark to determine CPU and memory requirements.

Figure 2-3 and Figure 2-4 show, for Teamcenter 10.1 and 11.2.1 respectively, the resource utilization for the sequences in Table 2-1 in a 2-tier deployment. This further illustrates how CPU / RAM use can vary significantly for each type of user or operation performed.

In these examples, red lines show the CPU utilization of Teamcenter related processes during the execution of the scenarios; blue lines show the average CPU for that user type. Green lines show the actual memory used[4], both shared and private.

For most user types the CPU utilization for this usage profile has decreased somewhat from Teamcenter 10.1 except for the 'ug0' type, which increased slightly in TC 11.2.1.

These charts show how resource requirements for specific sets of operations (i.e. the usage profile) can be substantially different. Resource requirements in your production environment are also likely to be different from those listed here. This highlights the need to use the sizing information in this section *as a guideline for pilot system deployments only*. Then measure workstation resource utilization of the pilot deployment using the actual operations expected to be executed by typical categories of users.

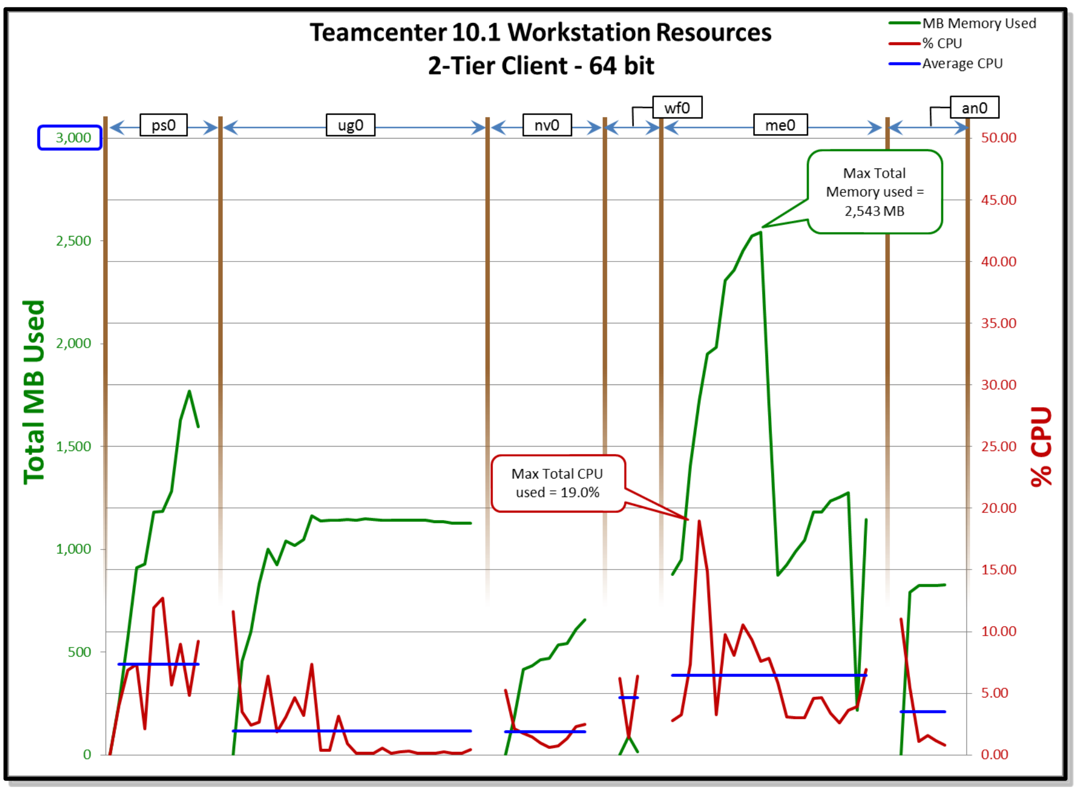


Figure 2-3, Tc 10.1 2-Tier Rich Client Resource Utilization (64 bit)[5]

---

[4] Actual memory used is measured by subtracting instantaneous memory available from that available at the start of the measurement.

[5] Showing memory on left in MB and CPU resources on right in percent of CPU.

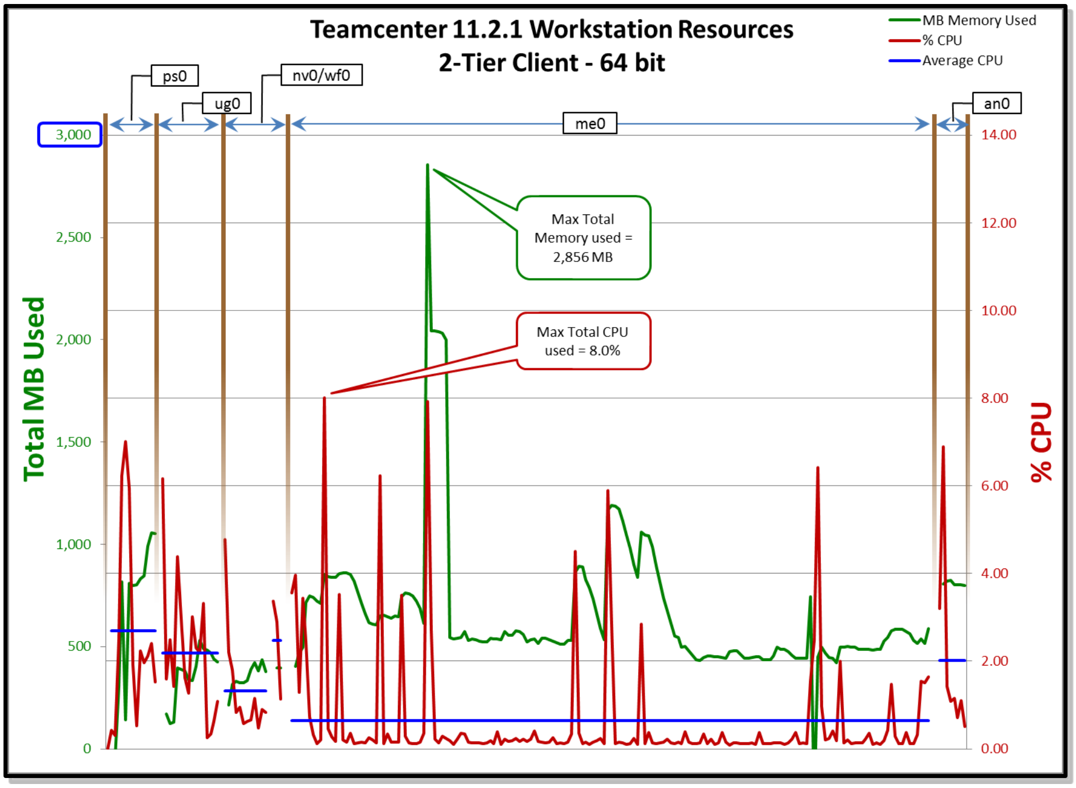


Figure 2-4, Tc 11.2.1 2-Tier Rich Client Resource Utilization (64 bit)[6]

[6] Showing memory on left in MB and CPU resources on right.

Figure 2-5 and Figure 2-6 show client utilization for the same usage profile for 4-tier deployments of 64-bit Teamcenter clients.

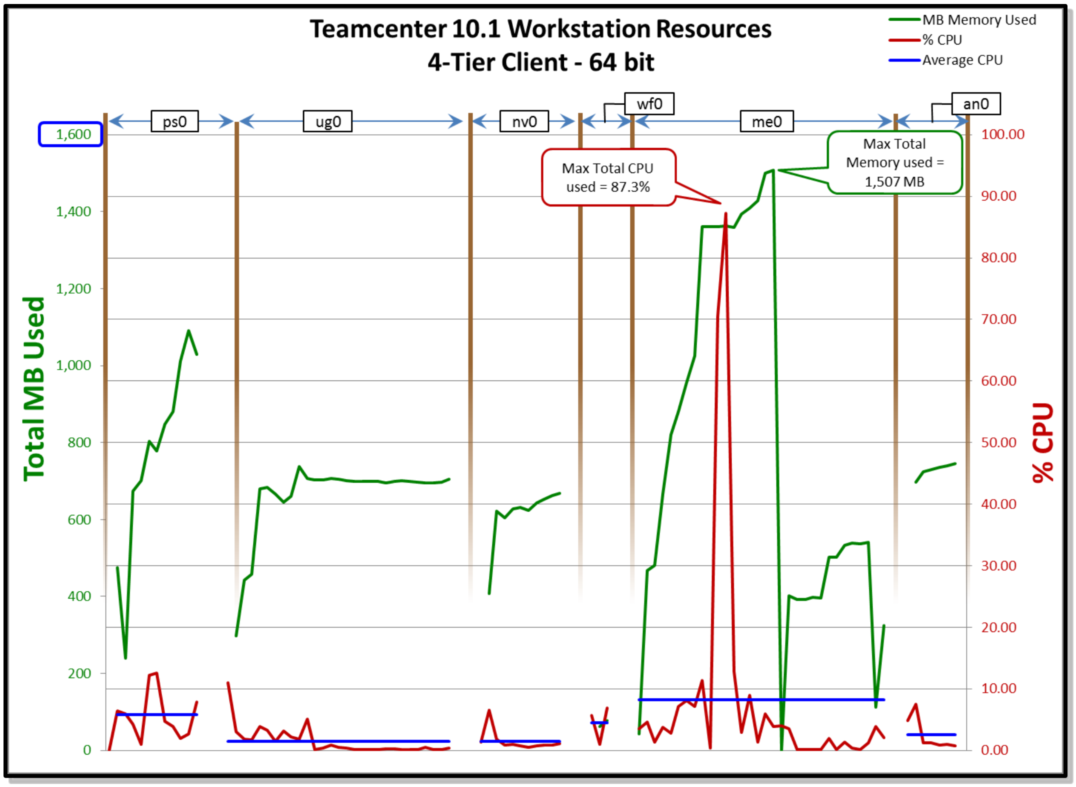


Figure 2-5, Tc 10.1 4-Tier Rich Client Resource Utilization (64 bit)[7]

[7] Showing memory on left in MB and CPU resources on right in percent of one CPU.

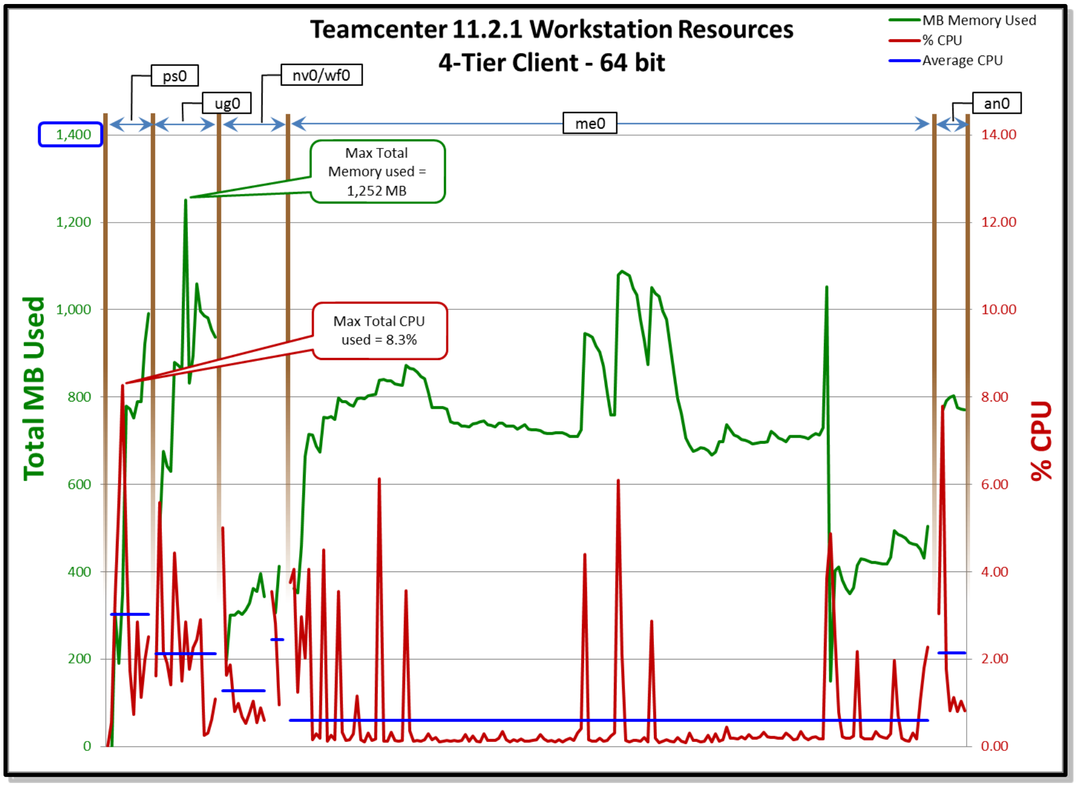


Figure 2-6, Tc 11.2.1 4-Tier Rich Client Resource Utilization (64 bit)[8]

For a 4-tier deployment, TC 11.2.1 CPU utilization for this usage profile decreased from Teamcenter 10.1 for all five types.

Note that some activities can consume a full CPU for long periods of time. If your usage profile includes long running transactions, monitor client CPU to determine if the any cores are 100%. If so, faster processors may reduce the time to execute those transactions.

To maximize workstation performance be certain to allocate sufficient physical RAM to contain the entire memory *Private Bytes* or *Working Set*[9] required for the operations used (i.e. the usage profile) in addition to the OS and any other applications that may be running on the machine.

The guidelines outlined in the following sections are based on these metrics from 2 and 4-tier benchmarks with 64 bit executables and can be used to estimate initial Rich Client workstation requirements. Compare the activities in Table 2-1, *Rich Client Benchmark Activities* with the activities anticipated your production environment and adjust sizing requirements accordingly. For example, if users typically view/edit product structures

---

[8] Showing memory on left in MB and CPU resources on right in percent of one CPU.

[9] Whichever is greater

significantly larger than 3000 components, they may require significantly more memory to ensure adequate performance.

Again, resource requirements for specific sets of operations used in your deployment (i.e. the usage profile) can be substantially different from those listed here. Use the sizing information in this section as a guideline for ***pilot system deployments only****. Then measure workstation resource utilization of the pilot deployment using the actual operations expected to be executed by typical categories of users.*

2.3.2 *Memory Requirements*

Table 2-2 shows the memory consumed by the activities in Table 2-1 for a typical and SM / MPP user.

Table 2-2, Single User Rich Client Workstation Memory Measurements

User Type		4-tier 64 bit	2-tier 64 bit
Typical (Navigator, My Worklist, Requirements…)			
	Memory Used	0.8 GB	1.2 GB
SM / MPP (Large Structures expanded / viewed in SM / MPP)			
	Memory Used	1.5 GB	2.6 GB

A single user 4-tier Rich Client workstation should be configured with a *minimum* of 4.0 GB of real memory not considering other applications running on the machine such as NX. 2-tier clients should contain a minimum of 6.0 GB. Client workstations will require a substantial amount of memory if massive assemblies are expanded fully in SM or viewed with the Rich Client Viewer.

Table 2-3, Memory Sizing Guidelines – Single User Client Workstation

Configuration	4-tier	2-tier
Minimum	4.0 GB	6.0 GB
Recommended	6.0 GB	8.0 GB

When in doubt, monitor memory utilization and memory paging activity in a production use while executing frequently used and/or long running transactions (e.g. with perfmon or Resource Monitor on Windows). Memory paging may indicate that additional memory is required.

2.3.3 CPU Requirements

Table 2-4 shows the CPU utilization for the activities in Table 2-1 for a typical, SM, and MPP user. Note the % CPU values listed here are estimates derived from those in Figure 2-4 and Figure 2-6.

The system used to execute this benchmark has an equivalent SPECint_rate2006 value of ~221. (*http://www.spec.org*)

- Intel Xeon E3-1270 v3, 3.5 GHz Quad-Core CPU

Table 2-4, Client SPECint_rate2006 Requirements

	4-tier, 64 bit		2-tier, 64 bit	
User Type	**% CPU**	**SiR**	**% CPU**	**SiR**
Typical	10%	22.1	12%	26.5
SM / MPP	40%	88.4	20%	44.2
Overall	**25.0%**	**55.3**	**16.0%**	**35.4**

Always add 25% more capacity to keep client CPU utilization below 80% to reduce the possibility of overloading the client machines. With that guideline, using the overall requirements above for examples:

- A single user 64 bit Rich Client workstation configured in 4-tier mode should be equipped with a SPECint_rate2006 value of at least 60 to provide some excess capacity, again without regard to other applications.
- A system configured in 2-tier mode with 64 bit Teamcenter should be equipped with a SPECint_rate2006 value of at least 40 to provide some excess capacity.

2.3.4 Network Considerations

Siemens PLM Software continues to improve the performance of Teamcenter clients in high latency Wide Area Networks (WANs). WAN performance is considered to be acceptable if the average response time of the APA standard performance usage profile (Table 2-1) does not exceed five times (5x) the average in a Gbit LAN network.

Siemens PLM Software supports 4-tier Rich Client performance to 300ms on a T1 link with Tc 8.3 and later. Depending on your usage profile and WAN characteristics, Tc 11.2.1 may perform acceptably at even higher latencies. Measurements of 4-tier Rich Client in simulated WAN environments show performance is somewhat slower in 11.2.1 compared to the 10.1 baseline Teamcenter version. The average of all 11.2.1 transactions measured at 300ms with T1 bandwidth is now about 6.6x LAN response times, vs. 4.8x in 10.1.

Measurements have shown that deployment of 2-tier clients over WAN results in very unacceptable performance. *Any client not on the same LAN as the Web and Enterprise tiers should be deployed as 4-tier.*

An excellent document with guidelines for Network Performance Tuning is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

See also section 10.2.2.4, *Network Resources*, for further information about improving network performance and the suitability of various Teamcenter deployment options for WAN deployments.

2.4 Other Considerations

There are a number of Teamcenter, Windows, and/or third party software parameters that may need to be adjusted for deployment or performance reasons.

2.4.1 *Teamcenter Rich Client 64 bit Performance*

If you are upgrading from a 32 bit version of Teamcenter, the 64 bit Rich Client may run out of Java memory[10], which might impact performance or cause Teamcenter to fail. You may need to increase the maximum Java heap size (-Xmx) for the JVM in which Teamcenter runs. For example, update teamcenter.ini in Portal Root and change -Xmx512m to –Xmx1024. See SFB-Teamcenter-6737 for further details.

2.5 Monitor Client System Usage

See section 10.1.3, *Monitoring Performance*, for important information about monitoring system utilization on client machines.

[10] due to the additional memory required in the 64 bit Rich Client

Chapter

3 Managing Web Server Deployments

Chapter

3 Managing Web Server Deployments

This chapter provides sample guidelines to manage one of the several commercially available web application servers compatible with Teamcenter.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- Server installation for: *Windows*, *UNIX/Linux*
- *Web Application Deployment* for Teamcenter
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

3.1 Teamcenter Multi-tier Design

Teamcenter is implemented as a 4-tier architecture with client, web application, enterprise (business logic), and resource layers. The resource layer includes database and data file services, among others. The enterprise tier comprises a configurable pool(s)[11] of Teamcenter C++ server processes (tcserver), and a server manager that manages the pool of the tcserver processes. The enterprise tier is a separate deployment from the Web tier and can be installed on a separate or the same physical server.

The http server is generally deployed as an integral component of the Teamcenter web tier as shown in Figure 3-1. This is either an IIS/.net Web Application Server (WAS) on Windows, or a fully J2EE compliant WAS with support for UNIX and Windows platforms including WebLogic, WebSphere, JBoss and others. The IIS/.net or J2EE web-tier handles communications between thin or rich clients and the pool of tcserver

[11] There may be more than one Business Logic pool of tcservers; one pool per pool network identity (pool-ID, IP, Port combination).

processes. Clients communicate with the IIS / J2EE server via http(s); the IIS / J2EE server communicates with the tcserver pools via .net for IIS or TCTP[12] for J2EE.

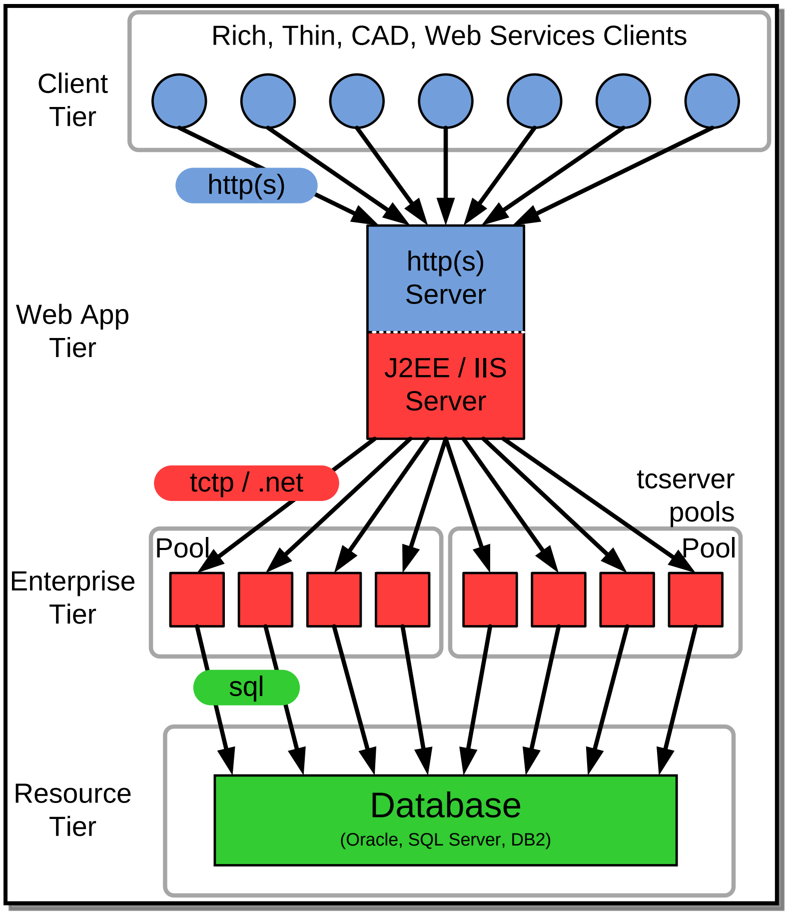


Figure 3-1, Teamcenter Architecture Overview

As Figure 3-1 shows, one or more tcserver pools, each coordinated by a *Server Pool Manager*, may be configured to serve the Web Application Server. For Rich Clients, each tcserver process in a pool contains state that ties it to a particular user, so that one tcserver cannot serve more than one client. Thin Clients keep no state information in the tcserver, however they do maintain affinity with a single tcserver process as long as it is available. Each *Server Pool Manager* maintains its respective pool, including the assignment state of each tcserver process, a number of pre-started 'warm' tcservers, how fast to start replacement tcservers, and how long to maintain tcservers no longer being used. See the chapter on *Enterprise Server Sizing Guidelines* elsewhere in this guide for more information about deploying *Server Pool Managers*.

[12] Teamcenter Transport Protocol, a PLM proprietary protocol, replaced CORBA IIOP in 11.1.

3.2 Teamcenter Web Application Servers

3.2.1 *J2EE Web Tier*

Teamcenter's implementation of J2EE Web Application Servers employs Teamcenter specific J2EE web applications (servlets) that are loaded into the J2EE server and configured at install time (using the **insweb** install program). It provides support for Rich, Thin, or Active Workspace Clients communicating via HTTP or HTTPS.

In Teamcenter's implementation of J2EE Web Application Servers, the HTTP and J2EE servers are normally run as a combined process. There is little advantage to separating them, although it is possible to configure an http proxy server between the J2EE server and the clients, or to configure firewalls.

3.2.2 *.NET Web Tier*

The Teamcenter .NET Web tier is an alternative to the Teamcenter J2EE Web tier for Microsoft Windows systems. The .NET Web tier supports four-tier Teamcenter deployments using Internet Information Services (IIS) as the application server. See .Net Web Tier Installation in the *Windows Server Installation* Guide from the *Teamcenter Help Collection*.

NOTE

This release of Teamcenter may require newer releases of some Web Application servers than previous releases of Teamcenter (e.g. WebLogic, WebSphere, IIS…). Consult the *Hardware and Software Certifications* page on GTAC for supported Web Application server versions.

In this document when a reference is made to "Web Application Server" or "J2EE Application Server", it includes both the J2EE specification "Web Container" and "EJB Container".

3.2.3 *Multiple Web Application Servers*

Although a single Web Application Server JVM has been shown to support several thousand concurrent sessions when properly tuned, it is possible to run multiple parallel J2EE servers for either load balancing and/or increased reliability. Typical deployments allow a single Web server host name (or IP address) and port to be specified by the clients, which will be routed to multiple host names / ports via software or hardware balancing.

There are also deployments that require multiple web server host names and/or ports. For example, if SSO and non-SSO access is required, separate web server and server pool stacks will have to be deployed as shown in Figure 3-2.

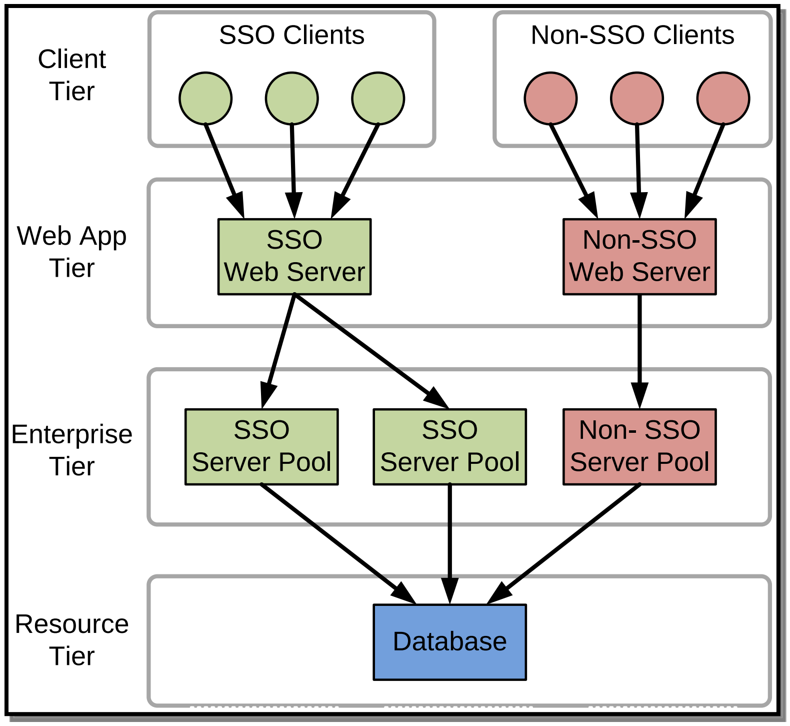


Figure 3-2, Multiple Web Servers – Multiple Hosts / Ports

3.2.3.1 Multiple Web Application Servers – Software Implementation

Web Application Servers may be clustered and/or load balanced to prevent single points of failure in the deployment.

Web app server(s) coordinate the *state* of tcserver assignments with the *Server Pool Manager* process(es) using a 'tree-cache' mechanism. The tree-cache ensures each client request is directed to its assigned tcserver process, regardless of which web app server handles the request. This tree-cache can be configured to use either peer-to-peer (the default) or multicast communications between the web app server and *Server Pool Manager*.

For example, the **insweb** tool supports building a proxy *.war file that can serve as the 'front-end' for several parallel, independent and/or clustered WebLogic servers in the background.

In the following example, if three managed servers were deployed as:

- Appsvr1 listening on machine:port **server1:7101**
- Appsvr2 listening on machine:port **server1:7201**
- Appsvr3 listening on machine:port **server1:7301**

Then a proxy war file can be configured to cycle requests 'round-robin' to the back-end servers by configuring a WebLogic HttpClusterServlet as follows:

```
<servlet>
    <servlet-name>HttpClusterServlet</servlet-name>
    <servlet-class>
        weblogic.servlet.proxy.HttpClusterServlet
    </servlet-class>
    <init-param>
        <param-name>WebLogicCluster</param-name>
        <param-value>server1:7101|server1:7201|server1:7301
        </param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>HttpClusterServlet</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

The resulting multi-server configuration is depicted in Figure 3-3:

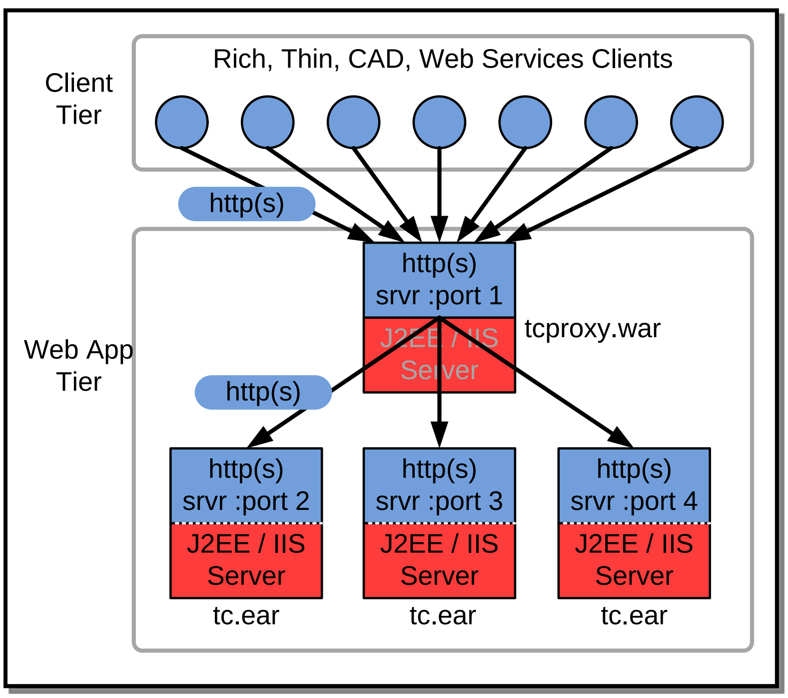


Figure 3-3, Multiple Web Servers – Software Load Balancing

The proxy servlet makes new assignments in a round-robin fashion to the back end servers. This can provide effective load balancing across multiple Java virtual machines while still providing for a single entry port, for configuration where the back end servers are not part of a server cluster.

Once a session is established on the back end server, clients are directed back to that server. If that server is in a cluster and the server stops responding, the client is directed to the secondary server in its cluster.

Keep in mind that the preceding is just one method of providing load balancing or failover capabilities.

3.2.3.2 Multiple Web Application Servers – Hardware Implementation

Alternatively, multiple web servers can be 'front-ended' by commercially available load balancing appliances.

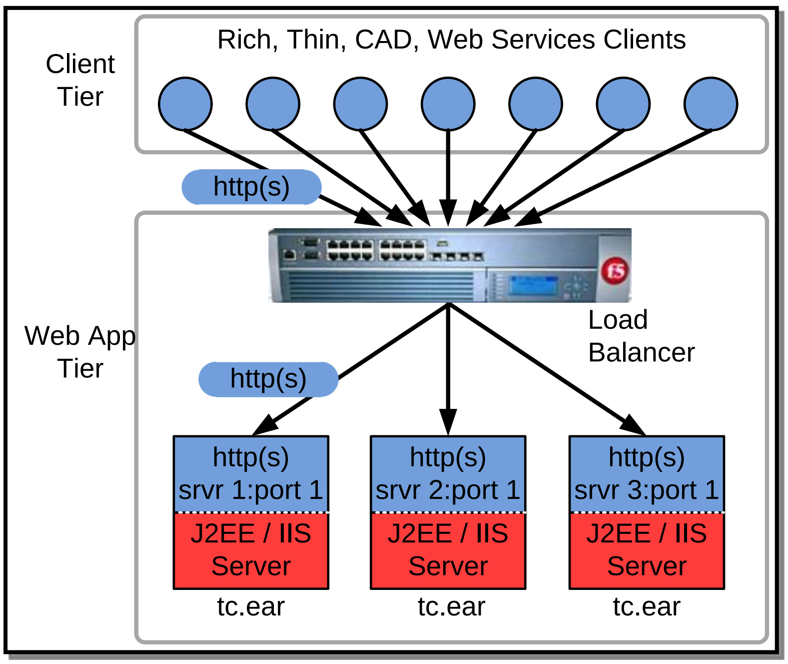


Figure 3-4, Multiple Web Servers – Hardware Load Balancing

A number of manufacturers offer load balancing appliances, each with their own deployment and configuration requirements. Consult the installation / configuration documentation for the specific appliance you plan to deploy.

3.3 Tuning the Web Server

Depending on your specific configuration, production deployments require certain minimal tuning to avoid performance and/or scalability issue[13]. Even minimal production deployments may require tuning of the Web Application Server and the server's Java Virtual Machine (JVM) for J2EE implementations. Larger production deployments may

[13] Proper tuning of a production Teamcenter environment includes the configuration of Oracle, the Server Pool Manager, as well as the J2EE Web Application Server.

require additional tuning of other Operating System configurations (e.g. TCP parameters, process file descriptor limits), and tcserver server manager configurations (e.g. number, location, tree-cache protocol).

Like most other tiers in a Teamcenter deployment, Web Server tuning is an iterative process – measure, adjust, and repeat. Also, tuning methods can vary somewhat depending the Web Application Server (WAS) deployed. Consult with the WAS provider for specific methods and recommendations. Many WAS providers post guidelines, tools, and other performance tuning resources on their product web sites. For example, at the time this revision was posted the following resources were available:

- IBM WebSphere: WebSphere Application Server Performance or WebSphere Application Server Performance Tuning Toolkit
- Oracle WebLogic: WebLogic Server Performance and Tuning
- Microsoft IIS: IIS Performance Tuning or Pump Up IIS Performance
- JBoss: Performance Tuning Guide [14]

3.3.1 *Tuning the J2EE Application Server*

For the Teamcenter Web tier, the documentation provided by the particular J2EE Application Server vendor[15] implemented should be reviewed by the system administrator for performance tuning advice.

Keep the following characteristics of the J2EE Application Server in mind when approaching the tuning of the Web Application Server:

- Parameters to tune a WebSphere J2EE Application Server will be different from those needed to tune a BEA WebLogic J2EE or any other Application Server.
- Its primary function is to manage connections to the tcserver pool(s) for various types of clients, and provide protocol translation. Its computing load is dominated by moving data. Hence the application should be tuned to optimize network I/O and the JVM should be tuned to optimize Garbage Collection (GC) of very transient memory.

3.3.1.1 Execute Pool Size

Nearly every J2EE Application Server has the concept of an execute pool size to serve client requests. BEA WebLogic Server release 9 and higher uses a self-tuned thread-pool so this is not a configuration issue for this application server implementation. The

[14] This is the latest JBoss Performance Tuning Guide available on the redhat documentation site at the time of this publication. Check also on the JBoss documentation index page for possible updates to this guide for later JBoss versions.

[15] WebLogic, WebSphere, JBoss, etc.

administrator should occasionally review if that pool is configured large enough to support the number of assigned *execution threads* to prevent large *execution queue delays*. As with other Teamcenter components, more is not always best. Since each *execution thread* typically results in an independent JVM thread, too many may overwhelm the system, while too few may result in user requests waiting for an available thread. However, since most of the processing time resulting from a user request is consumed in the tcserver on the Enterprise tier (or lower application processes), the number of assigned *execution threads* does not translate directly to increased load on the J2EE Application Server. Thus increasing the size of that pool is usually not detrimental if not beyond reason; several hundred may not be – several thousand might.

WebLogic Example: WebLogic supports a '`-Dweblogic.Chunksize`' parameter, documented by BEA to optimize certain I/O characteristics. Measurements with previous versions of Teamcenter have shown that a value of 16368 provides some benefit over the default value (References: "Bea WebLogic Server/WebLogic Server Performance and Tuning" version 10.0, March 30, 2007 "*Oracle® Fusion Middleware Performance and Tuning for Oracle WebLogic Server 11g Release 1 (10.3.1)*" → "Top Tuning Recommendations for WebLogic Server").

3.3.1.2 Tuning the Web Session Timeout

The Teamcenter Web Application Server maintains an active session per logged-in client. If this session times-out due to a period of inactivity, an active browser client would need to manually provide login information upon the next action request. In general, J2EE Application Servers commonly use a default of 30 minutes for session timeouts. To enhance the user experience for Teamcenter browser clients, it is recommended that this timeout be extended to something larger, such as in the range of 8 to 24 hours (480 to 1440 minutes). The performance impact of maintaining large numbers of idle sessions in the Teamcenter Web Application Server is minimal. Furthermore, an explicit logout will remove the session.

The timeout is controlled by the '**web-app -> session-config -> session-timeout**' element in the '**WEB-INF/web.xml**' file. That file is packaged within the '**tc.war**' file, which in turn is deployed within the '**tc.ear**' file during the install process:

```
<web-app>
…
  <session-config>
    <session-timeout>480</session-timeout>
  </session-config>
…
</web-app>
```

This timeout can be changed by modifying the session timeout during the install process and deploying the new '**tc.ear**', or by manually updating the value in the deployed '**tc.ear**' and then restarting the J2EE application server.

3.3.2 *JVM Tuning*

Teamcenter Web-tier J2EE application servers, regardless of web application, run within JVMs. (Note that IIS does not run in a JVM.) In fact, along with Teamcenter Web-tier application server, the managers for tcservers and FMS cache and Teamcenter rich clients run within JVMs. The JVM's memory management function, or garbage collection (GC), provides one of the biggest opportunities for improving JVM performance, thus improving Teamcenter performance. You can improve performance in production servers with proper configuration of JVM parameters, particularly those related to memory usage and garbage collection (GC).

JVM tuning is the process of adjusting the Java runtime parameters to improve performance. Default JVM options set at installation by J2EE applications (WebSphere, WebLogic, etc.) are generally not optimally set for production environments, but rather a single user or development setting. For example, there are two flavors of JVMs, 'server' and 'client'. If defaults are accepted during JVM installation, the "-client" settings will take effect when starting the JVM, which is more suitable for a single user. For the Web Application tier, configure for the 'server' JVM, which is designed for throughput.

In a Teamcenter deployment, the primary JVM to worry about is the one running the J2EE Web Application server. Large amounts of data returned to the web-tier from tcservers via IIOP are brought into the web tier before it can be sent to rich client. Garbage collection takes place on the JVM to reclaim the memory used to support the transient data previously transferred from tcserver. In a Teamcenter deployment, large amount of transient data due to high numbers of concurrent users can cause JVM garbage collection to happen more often and/or take longer. When garbage collection is executing JVM throughput is substantially reduced.

Two types of garbage collection can be encountered: 'full' and 'concurrent'. During full GC, the JVM is fully dedicated to running GC in one thread; all else stops in the JVM. While all activities in the JVM are halted during the full garbage collection, new requests from clients will be backing up in queue. Response times can be degraded due to higher simultaneous demand on the J2EE app server following a full GC when the JVM is catching up with the requests queued during the GC. It is possible another full GC may be triggered while it's trying to process the queue, which would lead to further performance degradation. In contrast, concurrent GCs do not stop other transactions and can scale to effectively use multiple CPUs, if present. The goal is to configure the JVM on the Teamcenter web-tier application server(s) to avoid a high number of expensive full garbage collections, and use concurrent GC instead.

When it comes to tuning applications, no recommendations will fit all customers and environments. In general, a baseline should be created which includes a well-defined test procedure that closely models your business needs (user usage profile, for example). Results from all tuning and application changes can then be compared to the baseline by rerunning the same established suite of tests. Once tested and validated, changes that result in performance improvements can then be promoted to the production system. Be sure to read and follow the GC tuning documentation for the JVM being used.

NOTE

The default JVM runtime options may change from JVM release to release (e.g. Java 1.4 vs. Java 1.5) or OS to OS (e.g. Java stack size on Solaris vs. Windows). Parameters to tune a JRockit JVM will be different from those needed to tune a Sun 1.5 JVM or an IBM JVM.

If you suspect that the JVM is contributing to poor client performance, use diagnostic tools to examine JVM events such as garbage collection (GC). Some tools available include:

- Visual GC (visualgc)
- jstat
- jconsole
- Collect GC log by adding '-verbose:gc -Xloggc:<path>verbose-gc.log' to the JVM startup script.

3.3.2.1 VisualGC

JVM monitoring tools such as "Visual GC" or "jconsole" provide the amount of time taken by garbage collection activities in the running JVM. Let's look at two examples.

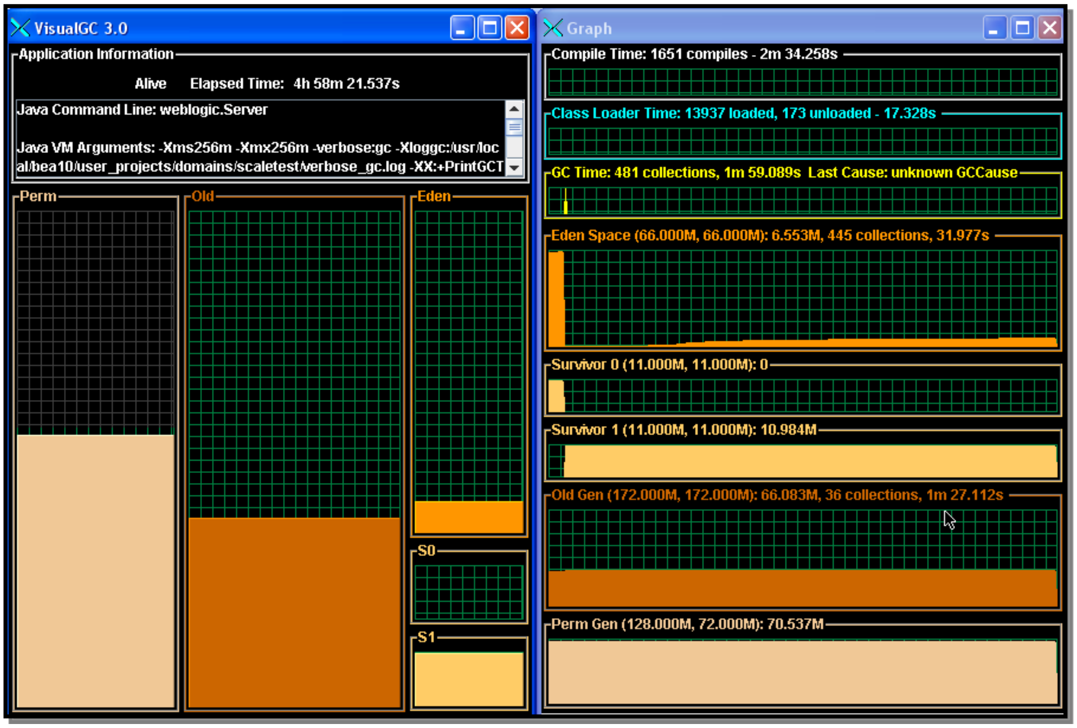


Figure 3-5, VisualGC Garbage Collection Activity (256 MB Heap Size)

In this example, shown in Figure 3-5, VisualGC shows the Web Server JVM during a scalability benchmark from the Siemens APA Lab. The JVM was started with "-server" mode and heap size of –Xms256 –Xmx256. 445 Young GC collections occurred from New Gen costing 32 seconds (Eden Space in the figure). Thirty-six full garbage collections (GCs) occurred from Old Gen costing 87 seconds. As state above, the goal is to avoid a high number of expensive full GCs. Adding heap space can sometimes reduce full GCs. In the following example min and max heap is increased to 1280 MB.

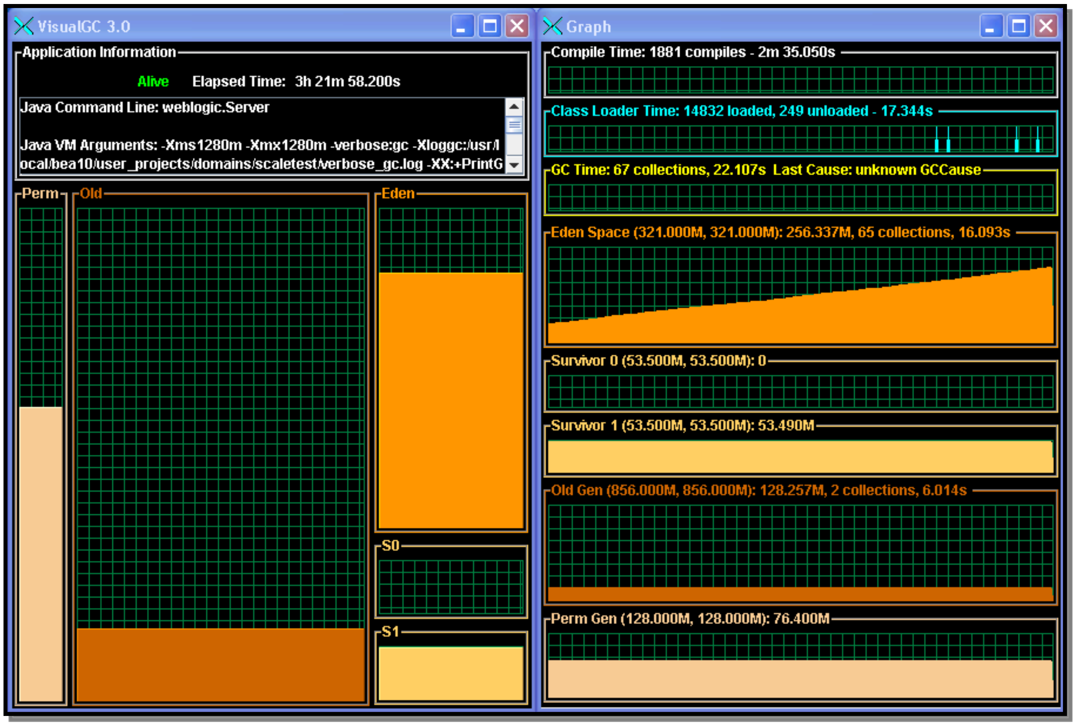


Figure 3-6, VisualGC Garbage Collection Activity (1.2 GB heap Size)

Figure 3-6 shows VisualGC monitoring a JVM during another scalability benchmark from the APA Lab. The JVM was again started with "-server" mode but this time with a heap size of –Xms1280 –Xmx1280 (larger heap size). In this example there were only 65 Young GCs occurred from New Gen (Eden Space) taking 16 seconds and only 2 full GCs occurred from Old Gen taking 6 seconds, a significant reduction resulting in improved Teamcenter performance.

Note too that if you have a very large number of Young GCs with a significant amount of memory allocated to heap, it can also take a long time to execute garbage collection. So it is generally better to use 'just enough' memory and not allocate too much.

3.3.2.2 Jstat

"jstat" is a command line tool that collects and logs performance statistics as specified by the command line options. Use "-gc" to show sizes for capacity and usage on each generation. "-qcutil" shows percentage of usage on each generation.

```
$JAVA_HOME/bin/jstat -gc 15046
 S0C      S1C      S0U     S1U       YGC      YGCT     FGC      FGCT      GCT
4096.0   4096.0   0.0     4088.8     341     15.349   11       26.941    42.290
```

NOTE

Do not directly compare time in Young (YGCT) to time in Full (FGCT). When Full GC occurs, all activities halt in the JVM. If system has 4 CPU 'threads', then one second in FGCT might as well be equal to 4 seconds, because during that time, only one CPU can be used by the entire JVM, and only for GC. Conversely, one second in YGCT might not delay any transaction if there were an idle CPU to run GC.

3.3.2.3 GC Log file

GC log files generated using the "-verbose:gc -Xloggc:logfile" option prints information at every collection. If you invoke the JVM log enabled, the garbage collector will write out what it's doing to the named log file. Each line in the log file corresponds to a GC operation. There are two types of lines, one for partial GC and one for full GC.

```
2432.956: [GC [PSYoungGen: 78830K->11255K(78848K)] 238820K->181445K(254976K), 0.0800776 secs]

2433.037: [Full GC [PSYoungGen: 11255K->0K(78848K)] [PSOldGen: 170190K->101671K(176128K)] 181445K->101671K(254976K) [PSPermGen: 72968K->72968K(131072K)], 2.2840980 secs]
```

"-XX:+PrintGCTimeStamps" option prints a time stamp on each GC entry (sec) since JVM startup. Example above shows at 2433 second into the JVM session, a Full GC occurred which took 2.2830980 secs.

"-XX:+PrintGCDetails" option tells the JVM to be verbose when printing out garbage collection data. Specifically, it not only tells you that there was a collection, but also what impact it had on the different generations. The example above shows statistics with all generations.

NOTE

Use the above options for logging GC in a test environment in your effort to analyze and tune the JVM. These options are not recommended to be applied as starting point for standard production system settings.

3.3.2.4 APA Lab Sun Example with JVM tuning using WebLogic v10

For the Sun JVM this means maximizing the size of the New (Eden) generation memory heap and maximizing Garbage Collection from that generation. This minimizes memory that becomes tenured, which requires more expensive Garbage Collection to recover.

The following is an example of parameters that should be considered for a Sun 1.5 JVM to optimize for high transient memory (several of which are specific to 1.5):

- **ms** and –**mx** affect overall JVM heap size (larger is generally better as long as the machine has the RAM for it)

- **XX:NewRatio** affects how much of overall heap is used to support the New generation
- **XX:SurvivorRatio** affects how much of New generation is reserved for survivors
- **XX:TargetSurvivorRatio** affects what threshold of the *New* generation survivors trigger the move to 'Tenured' (want to delay this so GC finds more 'New', hence set relatively high percentage).

For example, APA Lab Web-tier Application server has 32 Gig of RAM. The following parameters were configured for the JVM in WebLogic environment setup script (setDomainEnv.sh), and then scale benchmarks were executed to collect GC information:

```
MEM_ARGS="-Xms1280m –Xmx1280m -verbose:gc -
Xloggc:/usr/local/bea10/user_projects/domains/scaletest/ver
bose_gc_client.log -XX:+PrintGCTimeStamps -
XX:+PrintGCDetails -XX:+PrintTenuringDistribution -
XX:+DisableExplicitGC -XX:MaxPermSize=128m -Xss128k -XX:-
UseAdaptiveSizePolicy -XX:+UseParallelGC -
XX:SurvivorRatio=6"
```

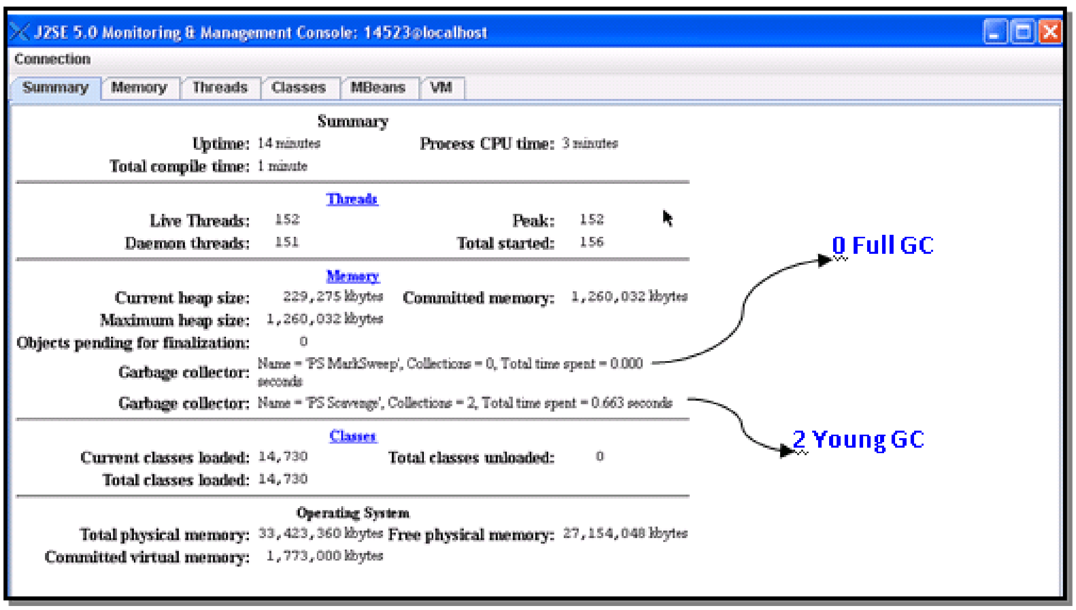


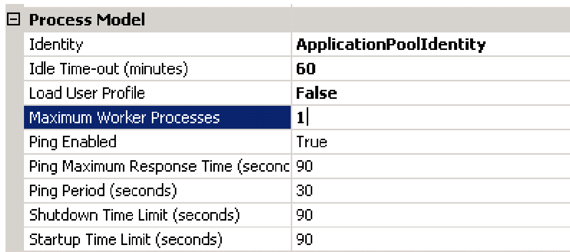
Figure 3-7, JConsole Monitoring and Management Console

3.3.3 *Tuning the IIS / .net Application Server*

For the Teamcenter Web tier, the IIS documentation provided by Microsoft should be reviewed by the system administrator for performance tuning advice.

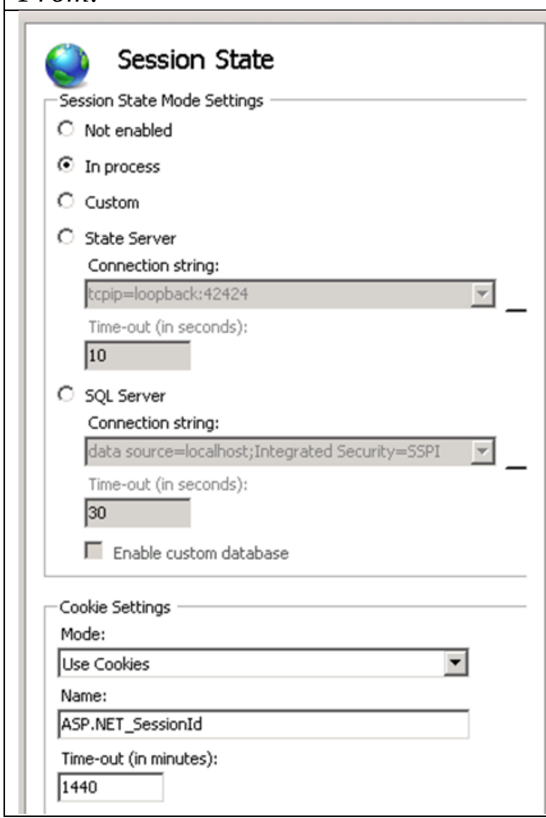
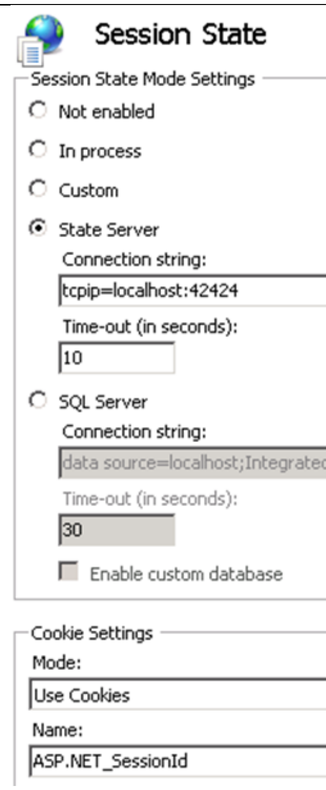
For very high loads or user counts IIS has the capability to run multiple 'worker' processes in what is called an 'IIS Web Garden' to handle requests in parallel.

If users experience transaction failures or performance delays, try creating a web garden for the Teamcenter ASP .NET Pool using the Internet Information Services (IIS) Manager. In the Actions pane, click Advanced Settings and update the value of Maximum Worker Processes to 2 or more.   Generally start with 3-5 worker processes and adjust as required.



3.3.3.1 Session State

To run with a Web Garden, the Session State service should also be enabled rather than using the In Process state. Select the Teamcenter Web Site and right mouse click on Session State. Select ‘State Server’ and update the Connection string only if required.

<table>
<tr><th>From:</th><th>To:</th></tr>
<tr><td>
</td>
<td>
</td></tr>
</table>

3.4 Web Application Server Sizing Guidelines

This section provides guidelines for establishing initial server configuration and sizing requirements. It provides information about the types of computing resource required for a specific sample usage profile[16], and aspects of each that may require adjustments to meet your unique usage requirements.

NOTE

Both the environments and usage profiles used to simulate Teamcenter operations, including scenarios, transactions, number and types of files, operating systems and databases have changed since the previous Teamcenter Deployment Guide was published. ***For these reasons it is not valid to compare estimates in this revision of the guide with previous revisions.*** Use the simulation information in this chapter and extrapolate from that to estimate the target pilot environment.

3.4.1 *Sizing Quick Reference*

This quick reference is to help locate Enterprise Server sizing information quickly for those who have already read this chapter. It assumes the reader has a good understanding of the factors that affect sizing information for both the Oracle and Enterprise servers and is familiar with terms defined later in the chapter (e.g. Oracle Demand Rate (ODR), Server Demand Rate (SDR), Usage Profiles, user categories, etc.).

The tables below provide guidelines for initial sizing estimates *of pilot deployments* only. Do not use this information until you have reviewed this section in its entirety.

NOTE

Starting with TC 11.2.0, usage profiles for Thin and Rich Client are significantly different. The resource requirements for Thin and Rich Client are therefore not compatible at any tier.

[16] as measured in the Teamcenter APA Scalability Lab

Table 3-1, Web Server Sizing Quick Reference

Component	Guideline
CPU	Assuming the usage profile matches APA Benchmarks;

Peak and average SDR per user for measured platforms is listed below as SPECint_rate2006 values:

		Thin Client		Rich Client	
Platform	**Web App Server**	**Peak SiR06 / User**	**Avg SiR06 / User**	**Peak SiR06 / User**	**Avg SiR06 / User**
AIX Power 7+	WebSphere 8.5.5.4	0.0072	0.0019	0.0062	0.0024
Solaris T5-2 / T7-2[17]	WebLogic 12.1.2.0	0.0018	0.0007	0.0043	0.0021
Suse Intel Core i3	JBoss 7.1.0	0.0020	0.0010	0.0037	0.0019
Windows Intel	IIS 8.5[18]	0.0041	0.0016	0.0070	0.0037

Therefore, multiply the # of users by the average SDR and factor in an Operating Range Reserve of 20% to handle the login rate.

For example, using 2000 Rich Client users on Solaris with WebLogic:

Step 1) 0.0043 x 2000 = 8.6

Step 2) 8.6 ÷ 80% = 10.75

Step 3) Select a Suse system with a SPECint_rate2006 rating of **10.75** or more.

[17] Thin Client measured on T5-2, Rich Client measured on T7-2

[18] Average of Oracle and SQL Server runs

Table 3-1, Web Server Sizing Quick Reference

Component	Guideline
Memory	Each *Concurrent* user consumes approximately the amount of RAM and SWAP listed below in Megabytes.

	Thin Client		Rich Client	
Platform	**MB RAM / User**	**MB SWAP / User**	**MB RAM / User**	**MB SWAP / User**
WebSphere 8.5.5.4	0.91[19]	-	0.38[20]	-
WebLogic 12.1.2.0.0	0.87	1.05[21]	0.82	0.62
JBoss 7.1.0.Final	0.83	-	0.55	-
IIS 8.5[22]	0.11	-	0.03	-

Note that Solaris pre-allocates SWAP space when a process is instantiated, even though no actual paging may occur.

3.4.1 *Usage Profile*

Customers routinely request sizing recommendations based purely on an estimate of the number of users they expect to access the system. Unfortunately, it is simply not possible to make an accurate recommendation based on number of users alone. Teamcenter Web Application server size will vary depending upon the anticipated *Usage Profile* (section **4.5.2**), server hardware platform, and Web server product.

The predominant system sizing consideration is which Teamcenter product features are used, and how often.

NOTE

You should obtain empirical system utilization data specific to your anticipated usage from your pilot deployment, and then extrapolate that to estimate the system resources needed for the production environment. A pilot implementation is highly recommended to obtain this empirical utilization data before sizing for production.

Refer to the description of the Rich Client usage profiles in section **4.5.2**, which was used to derive this sizing information.

[19] This is the actual RAM used with AME factor set to 2.0 (see section 4.5.3.2). Memory measured by SAR and other utilities will be twice this value.

[20] This is the actual RAM used with AME factor set to 2.0 (see section 4.5.3.2). Memory measured by SAR and other utilities will be twice this value.

[21] A Solaris bug in SAR on T5-2 Zones shows swap utilization for all zones running in the web server. Measurements in the non-zoned T7-2 systems (LDOMs) show the correct web tier swap utilization.

[22] Average of Oracle and SQL Server runs

3.4.1 ***Web Application Server Sizing***

Based upon measurements performed in the Teamcenter APA lab, it was observed that memory consumption of the supported Web Application Servers was generally below 2 GB for up to 3000 users. Specific usage varied across platforms and between different vendor's Web Application Servers, see Table 3-1, Web Server Sizing Quick Reference.

On all platforms measured, swap/pagefile utilization was under 2 GB given that a sufficient amount of memory was available for web server processes. Keep in mind that Solaris pre-allocates space in the swap file, while AIX, Suse, and Windows do not. Consequently AIX, Suse, and Windows show very little or no space used in the swap / paging file column of Table 3-1, so deltas don't apply.

The table illustrates per user resource consumption of the Web Application Servers hosted on various hardware / OS combinations, as measured in the Teamcenter APA performance and Scalability Lab with the standard APA usage profile. For comparative purposes, CPU utilization is represented in SPECint_rate_2006 (SiR) values per user.

Overall, even though some *per user* values show significant differences, *total* CPU, RAM, and SWAP requirements for Web tier have not changed significantly for Teamcenter 11.2.1 from 10.1 except for Suse Linux.

Table 3-2, Web Server CPU, Memory, and Swap Usage – Rich Client

Teamcenter Version	Platform	Web Application Server	Number of Users	Peak SiR06 / User	Avg SiR06 / User	MB RAM / User[23]	MB SWAP / User
Tc 10.1	AIX ORA	WebSphere 8.5.5.4	500	0.0062	0.0027	0.67	-
	Solaris ORA	WebLogic 12.1.2.0.0	2000	0.0043	0.0017	0.78	0.50
	Suse ORA	JBoss 7.1.0.Final	1500	0.0140	0.0058	4.34	-
	Windows	IIS 8.5[24]	2000	0.0068	0.0035	0.04	-
Tc 11.2.1	AIX ORA	WebSphere 8.0.0.1	500	0.0062	0.0024	0.38	-
	Solaris ORA	WebLogic 12.1.1.0	2000	0.0043	0.0021	0.82	0.62
	Suse ORA	JBoss 7.1.0	1500	0.0037	0.0019	0.55	-
	Windows	IIS 7.5[25]	3000	0.0070	0.0037	0.03	-
Deltas	AIX ORA	WebSphere 8.5.5.4			-12.0%	-42.9%	
	Solaris ORA	WebLogic 12.1.2.0.0			21.1%	5.4%	24.4%
	Suse ORA	JBoss 7.1.0.Final		-73.5%	-67.7%	-87.4%	
	Windows	IIS 8.5	1000	1.8%	6.6%	-15.7%	

3.4.1.1 Web Application Server Memory Sizing

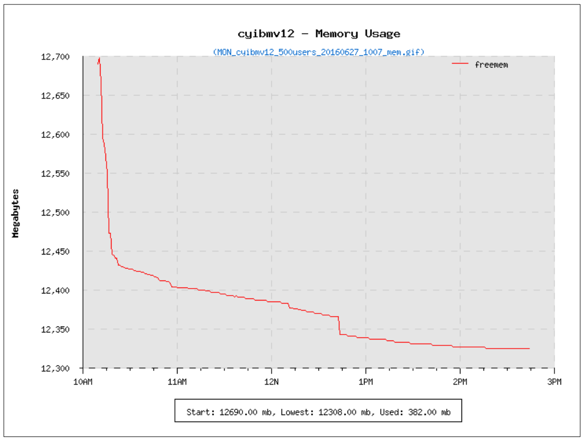
Memory consumption of the various web servers was found to be different across the measured operating system platforms. All tend to allocate RAM quickly and then maintain a level usage throughout the day as sessions are established and closed. The charts in Figure 3-8 track available memory as users login, work, and then logout (freemem on UNIX). As more users log in, more concurrent sessions are active, and more RAM is consumed. The memory listed in Table 3-1 is equal to the free memory before the measurement period starts, less the lowest amount observed (Used = Start - Lowest).

[23] For AIX, this is the actual RAM used with AME factor set to 2.0 (see section 4.5.3.2). Memory measured by SAR and other utilities will be twice this value.

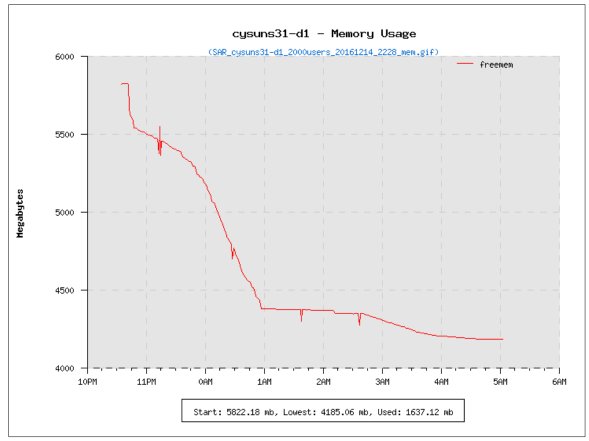
[24] Average of Oracle and SQL Server runs

[25] Average of Oracle and SQL Server runs

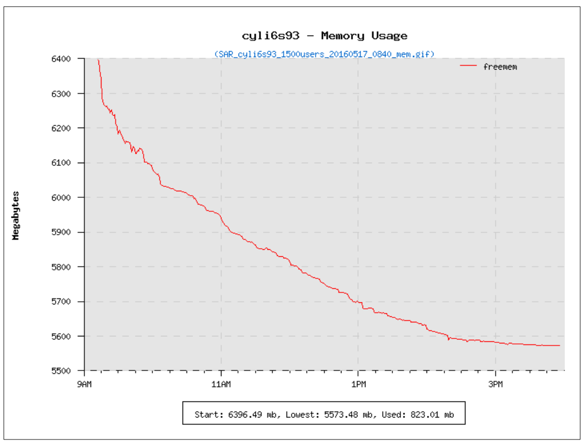
AIX WebSphere – 500 users[26]



Solaris WebLogic – 2000 users



Suse JBoss – 1500 users



Windows IIS/.net – 3000 users[27]

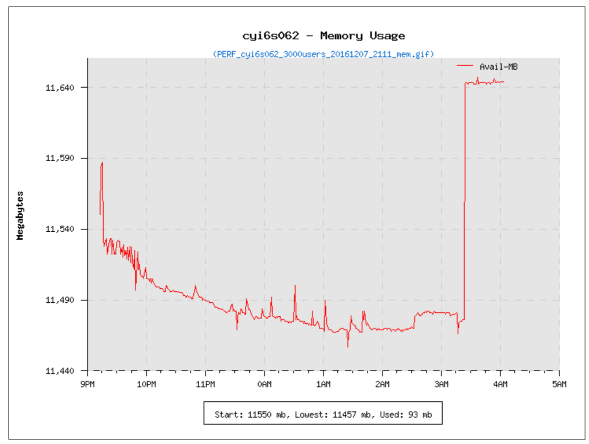


Figure 3-8, Rich Client Web Server RAM Utilization – All Platforms

3.4.1.2 Web Application Server Swapfile/Pagefile Sizing

With all user levels on the platforms measured, and given that a sufficient amount of memory was available, swap activity should be zero except on Solaris[28]. Swap utilization

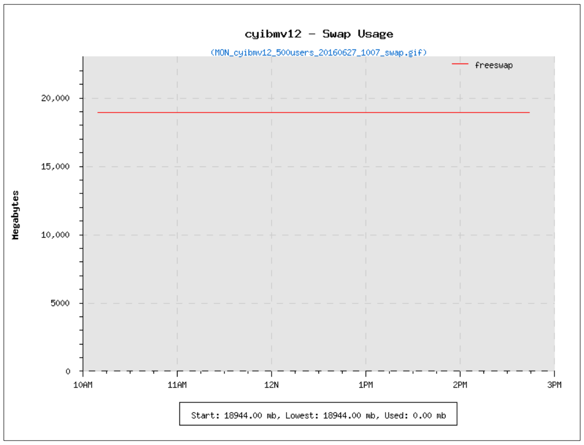
[26] This is the value of RAM measured with SAR. The actual RAM used with AME factor set to 2.0 will be half this value (see section 4.5.3.2).

[27] SQL Server example

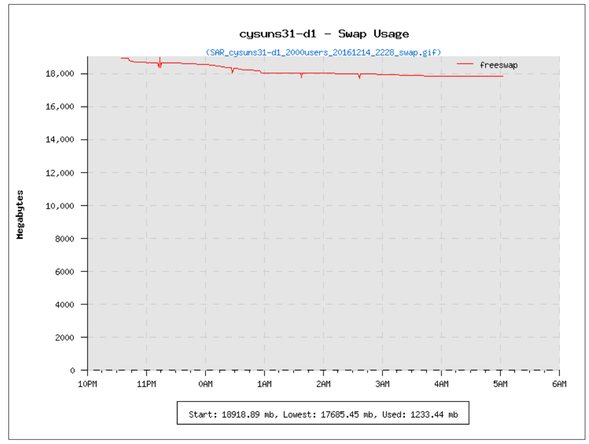
[28] Solaris pre-allocates space on the swap disk at process instantiation time.

on Solaris was under 2 GB. Note that UNIX charts show the amount of swap space available, while the Windows chart show percentage of the pagefile used.

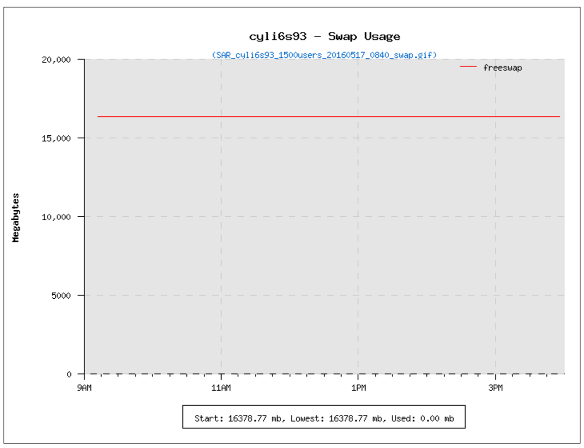
AIX WebSphere – 500 users



Solaris WebLogic – 2000 users



Suse JBoss – 1500 users



Windows IIS/.net – 3000 users

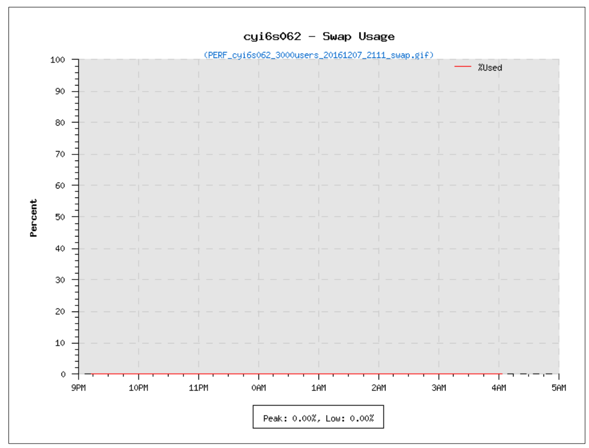


Figure 3-9, Rich Client Web Server SWAP / Pagefile Utilization – All Platforms

Comparing the baseline release to Teamcenter 11.2.1 shows nominal difference in average Web tier requirements across platforms except Suse, which was notably reduced.

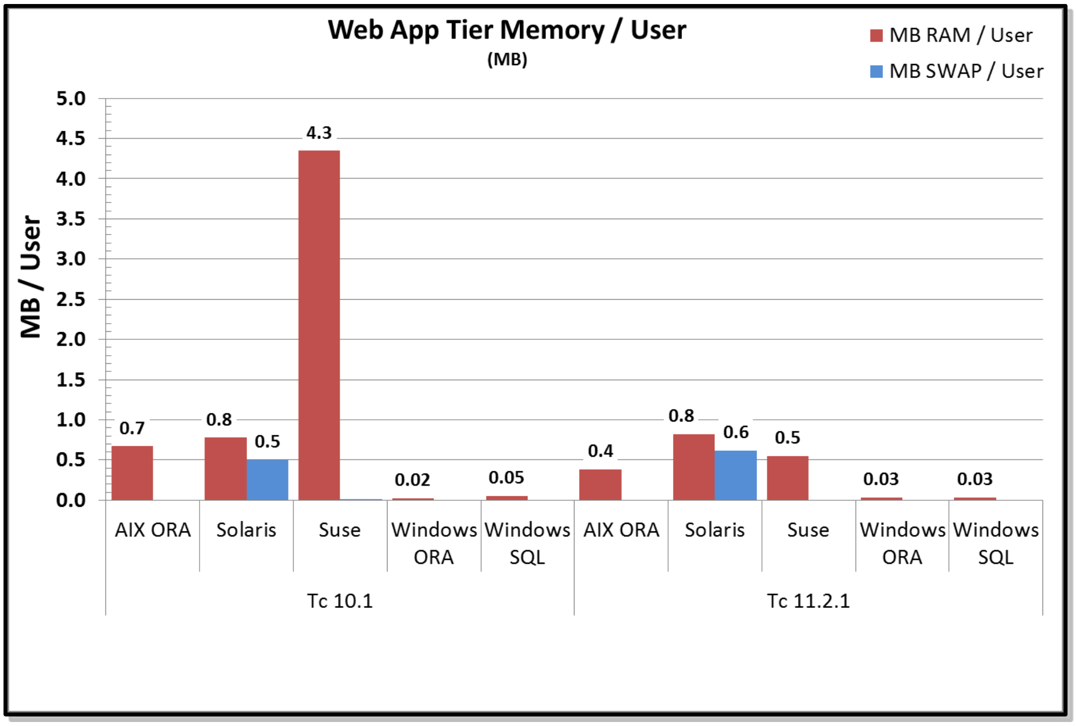
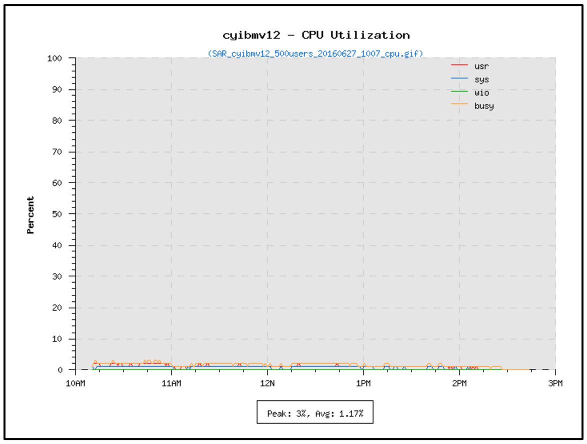


Figure 3-10, Rich Client Web Tier Platform per User Memory Utilization

3.4.1.3 Web Application Server CPU Sizing

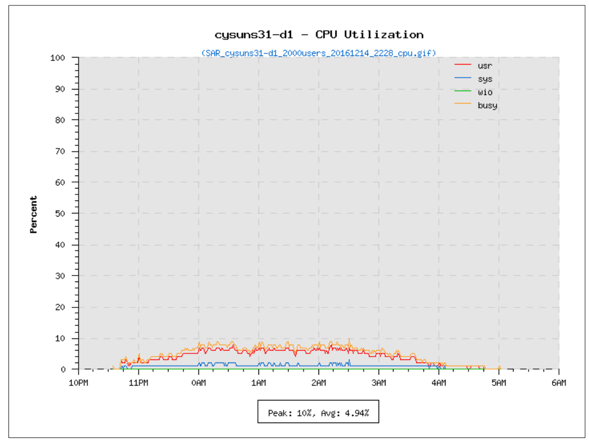
CPU utilization was noted to be minimal for usage scenarios up to 3000 users using various web servers of different platforms as listed in Table 3-1. Figure 3-11 shows CPU utilization during the course of a measurement period, along with number and type of CPU cores.

AIX WebSphere – 500 users



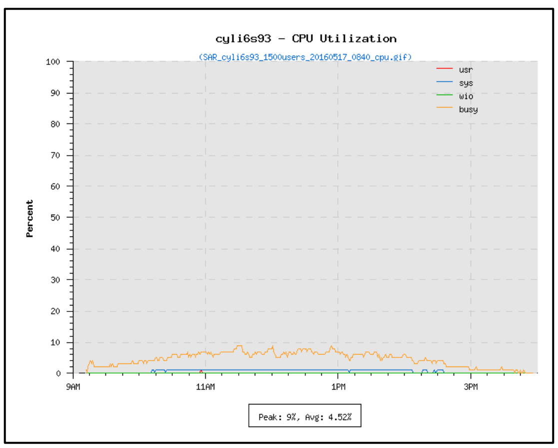
(2 x POWER 7+ @ 4116 MHz)

Solaris WebLogic – 2000 users



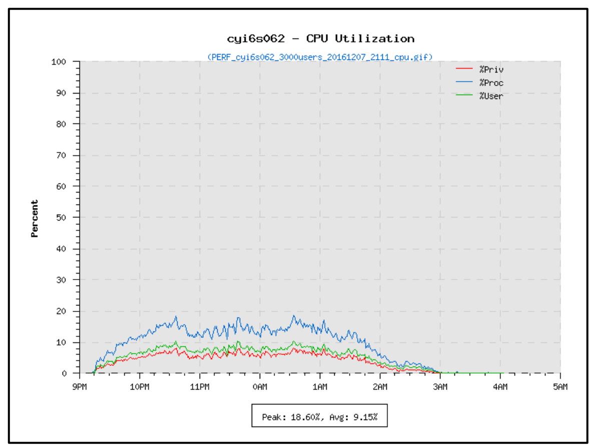
(2 x SPARC-T7 @ 4133 MHz)

Suse JBoss – 1500 users



(2 x Intel® Core® i3 CPU 540 @ 3.07GHz)

Windows IIS/.net – 3000 users[29]



(4 x Intel® Xeon® CPU X3460 @ 2.80GHz)

Figure 3-11, Rich Client Web Server CPU Utilization – All Platforms

[29] SQL Server example

Comparing the baseline release to Teamcenter 11.2.1 shows nominal difference or slight increase in average CPU requirements across platforms except for Suse, which shows a noticeable decrease in CPU utilized, as shown in Figure 3-12.

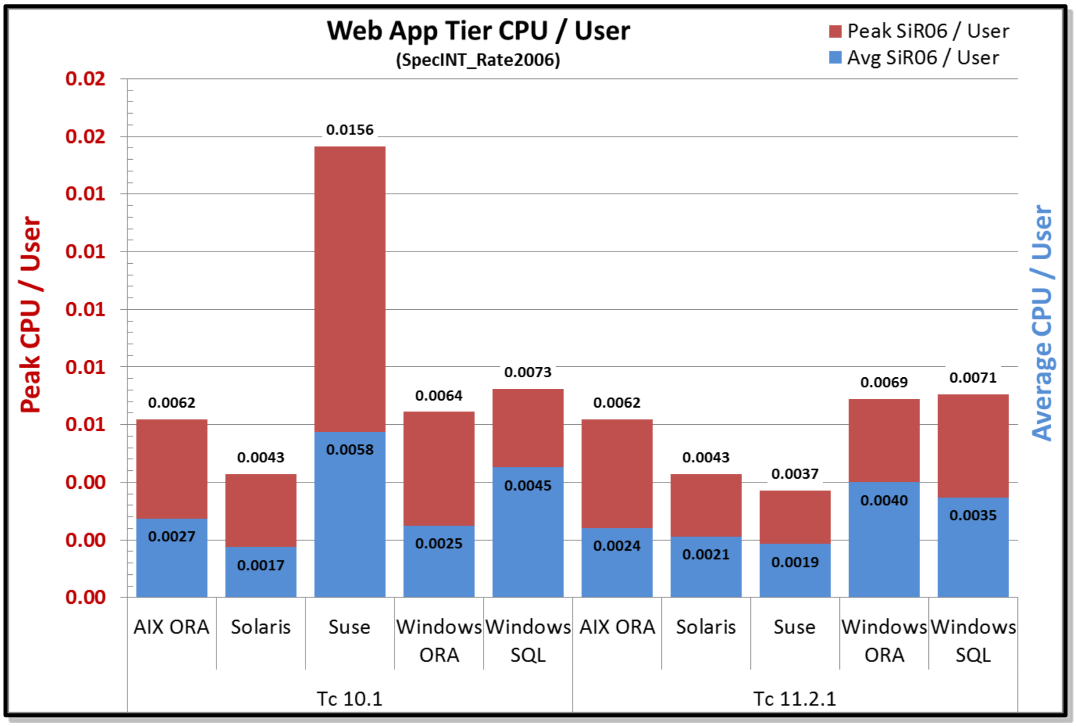


Figure 3-12, Rich Client Web Tier per User Platform CPU Utilization vs. Baseline

3.4.2 *Networking Considerations*

The Teamcenter web application deployed in the Web Server is configured by default to compress HTTP responses if the response size is larger than 500 bytes and the client can accept compressed results. The Rich Client and most browsers can accept compressed responses. Compression does not significantly increase or reduce the JVM Garbage Collection load. The compression functionality is configurable within the '**WEB-INF/web.xml**' file installed within the **tc.war** deployment. Note that the **tc.war** is normally packaged within the **tc.ear** deployment file.

The particular Operating System may need configuration tuning to support the network characteristics of the Web Application Server. For example, if you have configured the *Server Pool Manager* for high performance multicast protocol tree-cache, there may be network card configuration considerations. Refer to the *Troubleshooting four-tier architecture deployment* in the *Web Application Deployment* section of the *Installing Teamcenter* documentation.

See also section 10.2.2.4, *Network Resources*, for further information about improving network performance.

An excellent document with guidelines for *Network Performance Tuning* is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

The default HTTP mode for web-tier client connections is HTTP 1.1. Connections between the J2EE Application Server and tcserver process may come and go during the duration of any one tcserver, particularly if the maximum number of J2EE Application Server connections has been reached. Although the making and breaking of these connections is transparent to any given client request, machines may need TCP parameter tuning in order to optimize the availability and performance for very large numbers of connections to the machine.

For systems expected to support thousands of concurrent connections, there may also be a need to increase file descriptor limits.

3.5 Other Considerations

3.5.1 *BIOS settings*

Processor Power States

Some processors support multiple power states intended to conserve energy when that is important. If performance is critical to your deployment, consider disabling C-States (processor states) and / or P-States (performance states) in the BIOS settings. Not all processors support these options, and not all BIOS implementations support configuring them. Refer to the hardware user’s guide for your particular servers.

3.6 Monitor Web Server System Usage

See section 10.1.3, *Monitoring Performance*, for important information about monitoring system utilization on web server machines.

3.7 Teamcenter J2EE Application Response Time Logging (Optional)

For optimal performance of Teamcenter J2EE web app servers it is recommended to regularly obtain logs from the various components and examine them for trends or sudden changes in response times that may indicate correctable bottleneck. Logs can be obtained from the http server and the JVM, as well as the Teamcenter Application deployed in the J2EE Application Server.

The Teamcenter Application has a few strategically placed internal response-time instrumentation points. It is possible to configure the application to periodically log a summary of those response time statistics to a local file. See section 3.7.4, *Logged Elements* later in this chapter.

3.7.1 *HTTP Server Access Logs*

One standard logging point that can be used to record the load and response times of the Web Application Server are industry standard HTML access logs provided by the various HTTP servers. Most can be configured to report several parameters of interest for performance and usage analysis, e.g. time stamp, client IP address, URL called, returned bytes and response time. Consult the documentation for the particular HTTP server vendor used in the production system. This access log will have to be filtered to distinguish URL requests for static content, such as graphics files, as opposed to those URLs that invoke more dynamic processing of the deployed Teamcenter Application and hence invoke tcserver process functionality.

3.7.2 *JVM Performance Logging*

In some cases, depending upon the specific J2EE Application Server and the permissions granted to the deployed applications, it may be necessary to modify the J2EE Application Server's JVM classpath in order to enable performance sampling and logging. The sampling requires the Teamcenter Web Application to start its own sampling thread and to use JMX 1.2 classes.

3.7.3 *XML Configuration File for Response Time Logging*

The configuration of the response-time sampling and logging is controlled via a "**pref_exports.xml**" file placed in the startup directory of the J2EE server JVM. If modified, the configuration will take effect the next time the JVM is started. A "**pref_exports.xml**" file that follows later in this section is a recommended starting point for configuring such logging.

The applicable parameters in that file are formatted per the Java "preferences.dtd" rooted at the "user" node "com.teamcenter.mld.jmx". It is possible (through a parameter in that file) to open up an html adapter for a real-time view into the JMX MBeans that provide this monitoring and logging functionality. However the HTML server interface is for testing the JMX interfaces only, not for supporting production environments. The HTML adapter login and password can be set on the JVM startup line with parameters as follows:

```
-
Dcom.teamcenter.mld.HtmlServerLogin=changeme -Dcom.teamcent
er.mld.HtmlServerPassword=changeme"
```

The configuration file that follows will cause logging (at a 60-second granularity) of the various response time instrumentation points. You may wish to configure logging of longer duration filters at lower rates, e.g. hourly.

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE preferences SYSTEM
     'http://java.sun.com/dtd/preferences.dtd'>

<preferences EXTERNAL_XML_VERSION="1.0">
  <root type="user">
    <map />
    <node name="com">
      <map />
      <node name="teamcenter">
        <map />
        <node name="mld">
 <map />
 <node name="jmx">
   <map />
   <node name="com.teamcenter.mld">
     <map>
       <entry key="RunHtmlAdapter" value="false" />
       <entry key="HtmlAdapterPort" value="12345" />
     </map>
     <node name="com.teamcenter.mld.jmx.ResponseTimeMBean">
       <map>
<entry key="FilterAlgorithm" value="MA" />
<entry key="FilterMovingAvgTime" value="60000" />
       </map>
     </node>
     <node name="ResponseTimeSampler">
       <map>
<entry key="Active" value="true" />
<entry key="CdlFormat" value="true" />
<entry key="UseFiltered" value="true" />
<entry key="GranularityPeriod" value="60" />
       </map>
     </node>
     <node name="com.teamcenter.mld.jmx.RTGaugeMonitor">
       <map>
<entry key="HighThreshold" value="10000" />
<entry key="LowThreshold" value="8000" />
<entry key="NotifyHigh" value="false" />
<entry key="NotifyLow" value="false" />
       </map>
     </node>
   </node>
 </node>
        </node>
      </node>
    </node>
  </root>
</preferences>
```

3.7.4 *Logged Elements*

The above mentioned configurations will result in log entries placed in a log file "**RTEvents.txt**" file (by default) in that same directory.

The elements on each line of the resulting log file are:

- 'INFO' : the logging level
- A UTC date-timestamp in format 'yyyy/MM/dd-HH:mm:ss,SSS UTC', e.g. "2005/10/14-20:27:00,012 UTC"
- {*machine-name*}, e.g. "{xyzhost}"
- The Java class where instrumentation is implemented, e.g. "`com.teamcenter.presentation.wae.mld.MLDIntegrationService`"
- The name of the instrumented action, e.g. "invoke"
- The type of filter being logged: ("MA" – moving average; "NUM" – last 'N'; "C" – cumulative). "MA" is the most useful for periodic logging
- The response time statistics: Number of transactions in the filter span, average response time, minimum response time, maximum response time, standard deviation of response times, span of the filter, and the TPM (transactions per minute) rate. All times are in milliseconds. E.g. "23, avg, 771, 6, 5221, 1349, span, 60000, TPM, 23"

Since the log reports the Java class where instrumentation is implemented, it is helpful to know which class communicates with the client, which communicates with the *Server Pool Manager*, and which communicates with the tcserver process.

- clients =
  `com.teamcenter.presentation.wae.mld.MLDIntegrationService 'invoke' action`

- tcserver =
  `com.teamcenter.jeti.serveraccessor.ServerAccessor 'invoke' action`

- Server Pool Manager =
  `com.teamcenter.jeti.resourceadapter.spi.impl.ConnectionFactoryImpl 'getConnection' action`

The following shows a representative section of the "**RTEvents.txt**" file when sampling is enabled:

```
INFO - 2005/10/14-20:27:00,012 UTC {xyzhost}
   com.teamcenter.presentation.wae.mld.MLDIntegrationService, invoke,
      MA, 23, avg, 771, 6, 5221, 1349, span, 60000, TPM, 23

INFO - 2005/10/14-20:27:00,014 UTC {xyzhost}
   com.teamcenter.jeti.ejb.GatewayBean, invokeMethod,
      MA, 21, avg, 705, 26, 5116, 1262, span, 60000, TPM, 21

INFO - 2005/10/14-20:27:00,016 UTC {xyzhost}
   com.teamcenter.jeti.ejb.GatewayBean, authenticate,
      MA, 2, avg, 1335, 1331, 1339, 4, span, 60000, TPM, 2

INFO - 2005/10/14-20:27:00,018 UTC {xyzhost}
   com.teamcenter.jeti.resourceadapter.spi.impl.ConnectionFactoryImpl,
      getConnection,
      MA, 23, avg, 9, 3, 74, 17, span, 60000, TPM, 23

INFO - 2005/10/14-20:27:00,019 UTC {xyzhost}
   com.teamcenter.jeti.serveraccessor.ServerAccessor, invoke,
      MA, 21, avg, 692, 15, 5102, 1262, span, 60000, TPM, 21

INFO - 2005/10/14-20:27:00,024 UTC {xyzhost}
   com.teamcenter.jeti.serveraccessor.ServerAccessor, authenticate,
      MA, 2, avg, 1231, 1226, 1235, 4, span, 60000, TPM, 2

INFO - 2005/10/14-20:27:00,025 UTC {xyzhost}
   com.teamcenter.jeti.resourceadapter.spi.impl.ConnectionImpl,
      invokeMethod,
      MA, 21, avg, 699, 21, 5111, 1262, span, 60000, TPM, 21

INFO - 2005/10/14-20:27:00,027 UTC {xyzhost}
   com.teamcenter.jeti.resourceadapter.spi.impl.ConnectionImpl,
      authenticate,
      MA, 2, avg, 1268, 1264, 1272, 4, span, 60000, TPM, 2
```

This log file can be used to analyze historical transaction rates or to document load and response-time profiles of the Teamcenter Application deployment within the J2EE Application Server.

As discussed earlier, some J2EE Application Servers may require a modification to the JVM classpath entry. In some cases this extends to include the jar with com.teamcenter.mld.* package classes so that periodic response time sampling and logging will function properly. The jar file "plmJMXmonitor.jar" with these classes can be extracted from the tc.ear file, in the same manner as extracting the JMX 1.2 reference implementation.

3.8 Teamcenter and Firewalls

3.8.1 *Introduction*

This section outlines how to operate Teamcenter through a Firewall, what the possibilities are, and the limitations. Beginning with Tc 2005, commercial Web Application servers, which are more Firewall friendly, are now an integral part of the Teamcenter architecture.

Firewalls are secure gateways that control traffic into and out of a company's internal network, usually with multiple network interfaces. They are a generally a combination of hardware and software. In a typical configuration, one network interface on a '*gateway*' server connects to the public network and the other connects to the company's internal (or private) network. There are basically two different implementation approaches employed by leading firewall vendors:

3.8.1.1 IP Filtering:

This approach operates by blocking or allowing communication between networks or specific machines based solely on information contained in the IP packet headers, for example node address and/or port (socket) number.

3.8.1.2 Application Proxy:

In this approach, information flows through the firewall, but no outside packets do. Direct communication between the inside and the outside is severed, and firewall application(s) on the gateway act(s) as a data relay between inside and outside hosts according to the security policy.

3.8.2 *Teamcenter Firewall Configurations*

Three basic Firewall configurations are supported by Teamcenter as shown in the diagrams below. The configurations shown in Figure 3-13 and Figure 3-14 are the legacy configurations supported from earlier Teamcenter versions and are still supported for Teamcenter. The configuration shown in Figure 3-15 is supported in Teamcenter and is the future direction for Teamcenter WAN deployments.

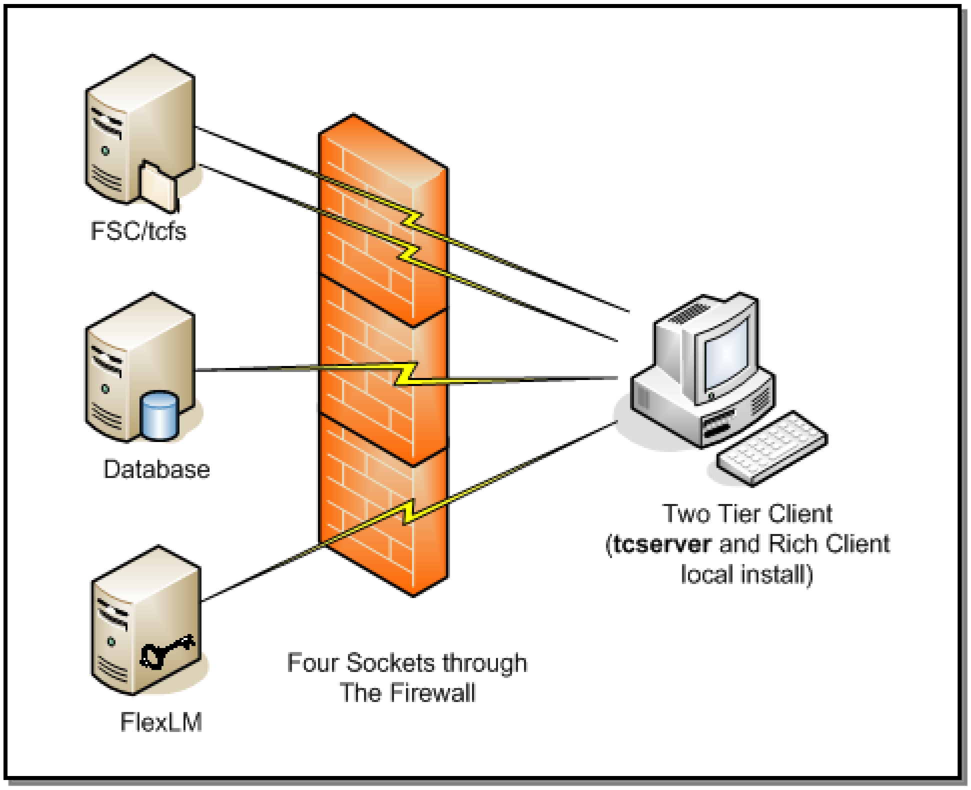


Figure 3-13, Firewall between 2-Tier Rich Clients and Data Servers

In Figure 3-13 the traffic passing through the firewall is:

1. OracleNet
2. FMS – socket Traffic
3. License traffic to FlexLM.

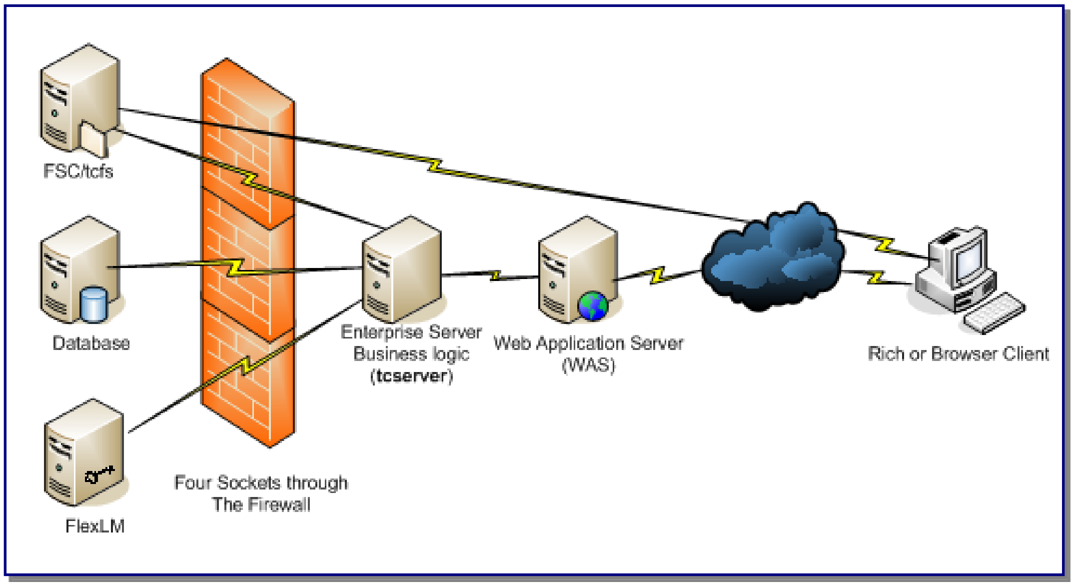


Figure 3-14, Firewall between Enterprise Server and Data Servers

In Figure 3-14 the traffic passing through the firewall is:

1. OracleNet
2. FMS – socket Traffic
3. License traffic to FlexLM.

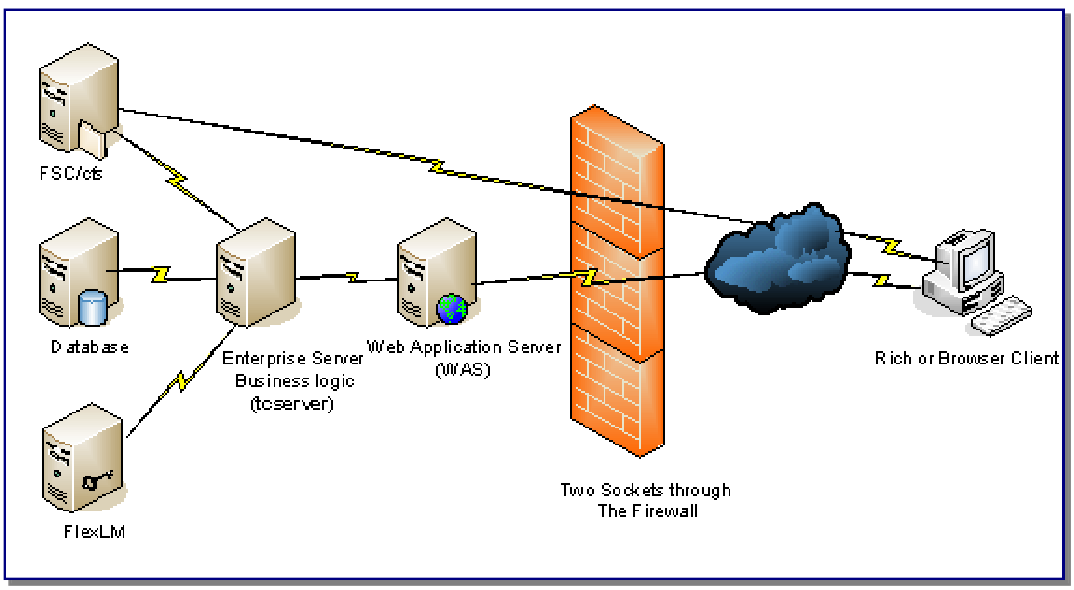


Figure 3-15, Firewall between Rich/Thin Clients and Web Application Server

In Figure 3-15 the traffic passing through the firewall is:

1. HTTP between Client and WAS
2. HTTP between Client and FSC

3.8.3 *Teamcenter Component Operation through a Firewall*

3.8.3.1 Web Client Considerations

Both the Thin and 4 tier Rich clients communicate using the HTTP (or HTTPS) protocol. HTTP and HTTPS operate on named sockets and work well with Firewalls.

3.8.3.2 Oracle Considerations

OracleNet on a UNIX server

In dedicated server mode, UNIX natively implements TCP/IP socket sharing, consequently all traffic passes through the name socket(s) defined in **TNSNAMES.ora** and **LISTENER.ora**. For example socket/port 1521.

If the system is configured to use MTS (Multi-Threaded Server), then MTS must be configured to specific TCP/IP ports. For example:

```
dispatchers="(address=(protocol=tcp)(host=xxx)(port=2700))
(dispatchers=1)"

dispatchers="(address=(protocol=tcp)(host=yyy)(port=2800))
(dispatchers=1)"
```

In this example, the firewall system must be configured to open TCP/IP ports 2700 and 2800.

Alternatively a system using MTS can install and configure Connection Manager (CMAN) to address this issue.

With a Firewall providing NAT (Network Address Translation) service, dedicated servers will work without a problem however MTS is only supported via CMAN using NAT.

Sqlnet/Net8 on Windows Servers

Unlike Unix Windows does not implement TCP/IP port sharing by default.

If using TCPS protocol or SSL, port re-direction will occur. The only solution is to use firewall vendor that supports Sqlnet/Net8 connectivity or configure Multi-Threaded Server (MTS).

For Oracle 10g and later, there is no longer redirection of the ports on Windows servers. Therefore, the setting of **USE_SHARED_SOCKET** is not required.

As with UNIX, an alternative is to use CMAN.

MTS require the setting and the socket definition shown above under UNIX servers. Dedicated servers will work without a problem with a Firewall using NAT (Network Address Translation). However MTS can only be supported via CMAN.

Also note that if using TCPS protocol or SSL, port re-direction will occur. The only solutions are to use a firewall vendor that supports Sqlnet/Net8 connectivity or configure Multi-Threaded Server (MTS).

3.8.3.3 FMS Considerations

FMS operates over named sockets. The default socket for FMS is 4445.

In general socket mode is recommended for Firewall use. NFS is not generally firewall friendly since it is based on RPCs which allocate sockets dynamically. However some platforms, SUN for example, do support named sockets for NFS to make it more firewall friendly. Also some Firewalls provide direct support for RPC. Consult your firewall provider.

3.8.3.4 FlexLM Considerations

Siemens PLM Software uses Globe Trotter's Flex license system. By default, the license delivered by Siemens PLM Software uses two sockets, one is named (usually 28000 by default), and the other is a dynamic connection to ugslmd.

To support Firewalls the dynamic socket allocation may be disabled by adding the named socket to the Flex licenses file. The entry:

```
VENDOR ugslmd
```

needs to be changed to:

```
VENDOR ugslmd port=28010
```

(or whatever available port/socket you want to use.)

For example:

```
SERVER cbrltpc12 ANY 28000
VENDOR ugslmd port 28010
############################################################
#   PACKAGE Lines of Bundle Suites       #
############################################################
```

3.8.4 Firewall Built-in Functionality

As noted above Firewall systems often have support for some protocols already built in for specific applications. Functionality that may be available includes:

- Oracle Net proxy functionality is provided by several vendors and this can be used with Teamcenter Oracle databases
- IIOP support

Consult your firewall provider for specific capabilities.

3.8.4.1 Firewall Configuration Issues

Setting the ‘IDLE Timeout’ value too low on the Firewall (e.g. 1 minute) can cause dead locks in the database. The timeout should be set to be longer than the expected idle period, typically 60 minutes.

When a user is working in a Teamcenter encapsulated application for a long period, NX for example, the Oracle connection between the tcserver and the database will be idle for a long period of time and may be closed by the Firewall. When the user saves or performs another action requiring database access the Teamcenter server process will reopen the connections. However if the ‘clearlocks’ utility is run while the database connection was closed by the firewall the user’s session will be read-only after reconnecting. Running clearlocks as a ‘cron’ or batch job on a system using Firewalls during the working day is therefore not recommend.

The Flex license connections use a heartbeat (keep-alive) to keep it apparently busy, therefore these will not be affected by Firewall Timeout periods.

3.9 Teamcenter and Proxy Servers

3.9.1 *Proxy Servers*

In computer networks, a ‘proxy server’ is a computer system or application program that acts as an intermediary for requests from clients seeking resources from other servers. An internet client connects to the proxy server requesting some service, such as a file or web page. The proxy server evaluates the request according to its security filtering rules and if validated, requests the service from the source intranet server on behalf of the client, then provides the resource to the client. This allows controlled, secure access to select machines or services, without providing access to the entire intranet.

Proxy servers generically come in two basic forms, Reverse proxy and Forward proxy. The term ‘proxy server’ usually refers to a *forward* proxy, similar to those used by organizations to connect their internal corporate network to the external internet.

A Reverse proxy is just a machine somewhere in the network, with one leg in one network, and the other leg in another network. In simple terms, a Reverse proxy will be placed where a corporate network is connected to the Internet or between two corporate Intranets, so that browsers in the Extranet[30] can talk to the proxy, which will forward all requests to the internal server for answering. Typically an Internet facing Reverse proxy hides the internal network from the outside world.

The Reverse Proxy server often performs other task such as load-balancing, authentication, decryption or caching. This leads to these proxy servers being identified as Firewalls or load balancers etc.

A Forward proxy is very similar, it is again a machine the connects the corporate intranet network to the internet, but this time so that browsers inside the corporate network can talk to the Forward proxy which will forward requests to the Internet based servers for answering. Besides forwarding requests it might also provide other functions such as content caching and filtering.

3.9.1.1 URL General Concepts

In order to discuss the configuration of proxy servers it is necessary to understand some basic concepts.

[30] An extranet is a computer network that allows controlled access to a company’s intranet from the internet, for specific purposes. An extranet can be viewed as an extension of a company's intranet that is extended to users outside the company (partners, vendors, suppliers…)

A URL[31] for a page is written in the form:

`http(s)://`*`server.domain.ext:port/URL`*

After the HTML for the page is loaded, all other URLs on the page itself should be server-relative, that is they will default to use the server the page was loaded from. They are either of the form *`/URL2`* or just *`URL3`* or `../`*`URL3`*. The written URLs within the HTML code should not contain a "protocol identifier" (`http(s)://` part) and ideally no "resource name" (server/domain name references).

The first important aspect to understand is that each and every request from the browser is always server-absolute. What it means is that each and every incoming HTTP request is triggered by a URL of the form:

`http(s)://`*`server.domain.ext:port/URL`*

That is, with protocol, server, port and server-relative URL. (Note: if unspecified the port has the default value of 80 for HTTP or 443 for HTTPS.)

So, if the URLs within HTML code do not contain protocol identifiers nor resource name, how does the browser build up the correct URLs? The answer is that the browser will always take the missing parts of information from the URL of the current page that is displayed. In the case of the absolute URL shown above, the browser will take the current active protocol and server information, and generate a new server-absolute URL:

`http(s)://`*`server.domain.ext:port/URL2`*

In the case of a relative URL, e.g. *`URL3`*, the browser will again use the information from the current loaded page to construct a server-absolute URL:

`http(s)://`*`server.domain.ext:port/URL3`*

What we notice is that as long as URLs in HTML pages never contain a "protocol identifier" (`http(s)://`) or any form of "resource name" (*`server.domain.ext:port`*), URL generation will always work correctly, independent of the actual protocol in use. This is why most code will work correctly in context of a proxy.

However if the server has issued an HTTP redirect, which can occur for many reasons, then the standard requires the use of absolute URLs.

---

[31] Uniform Resource Locator

3.9.2 Reverse Proxy Servers

3.9.2.1 Reverse Proxy Configurations

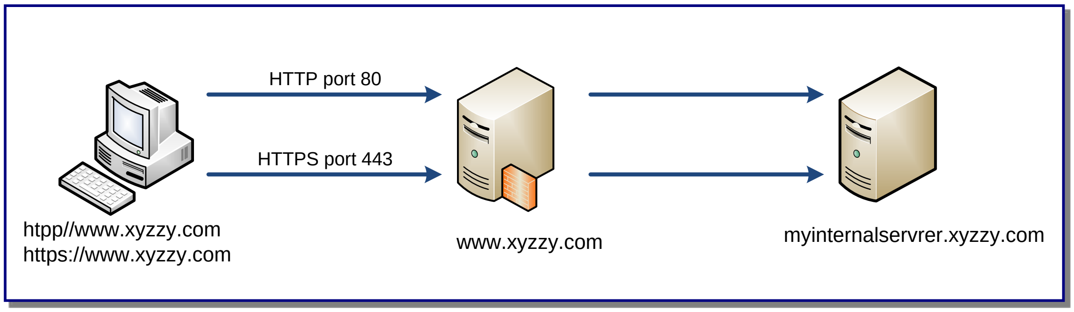


Figure 3-16, Basic Proxy Configuration

In the first scenario (*Basic Proxy Configuration*) a relatively simple configuration is considered, where the incoming ports used on the proxy matches exactly those on the server. So if we wish to make a protocol switch to HTTPS, we can take the name that the browser uses to talk to us (see section 3.9.2.2, *HTTP Header* for more detail) and then just look at our own port number for HTTPS.

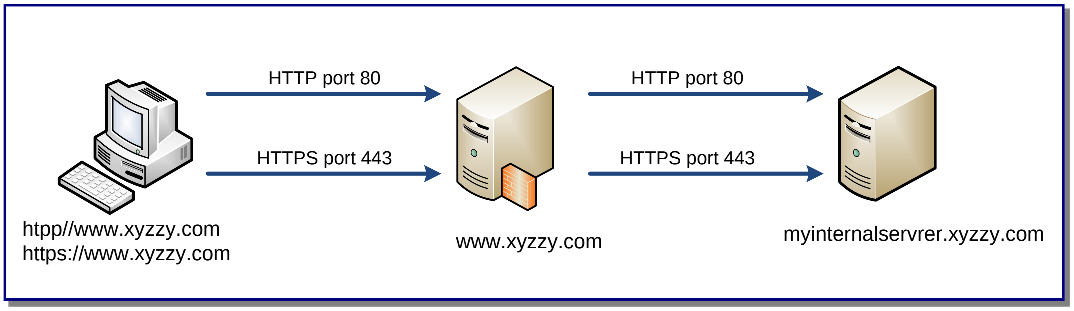


Figure 3-17, Advanced Proxy Configuration

In the second scenario (*Advanced Proxy Configuration*), already the naive approach fails. Here we have port mapping active. The browser talks to the proxy on the default ports (80 for HTTP and 443 for HTTPS). However, the proxy forwards the requests to the internal server on different ports. If protocol switching from HTTPS is not required, the server does not know the port number to use! It can only extract the server name from the Host: header, but has no immediate clue as to what ports are active on the proxy. Here, already, configuration data becomes critical.

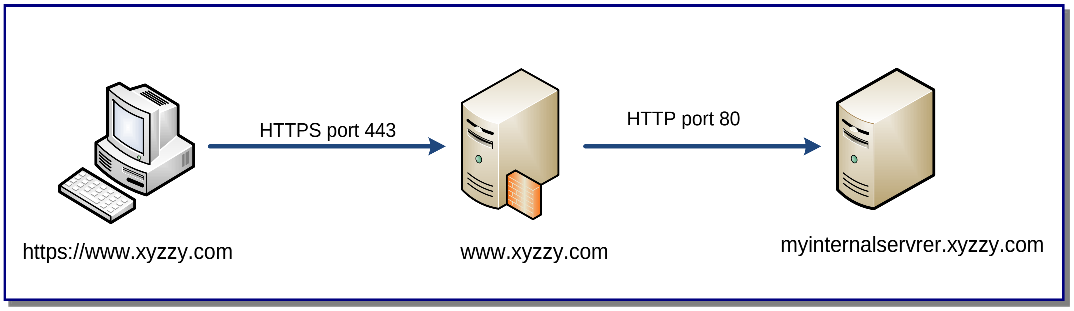


Figure 3-18, Protocol Switching

In the third scenario (*Protocol Switching*), we see in addition, a protocol switch. On the Internet, HTTPS is used to encrypt all data and ensure a secure transfer to your corporate network. However, once inside the controlled network, normal HTTP is used from the proxy to the server. The usual reason for not using HTTPS in the data center is it adds complexity for relatively little additional security. A typical problem is the Web Server is creating a URL based on the protocol used on its connection, i.e. HTTP rather than HTTPS. This mainly occurs when an absolute URL is used for redirects etc.

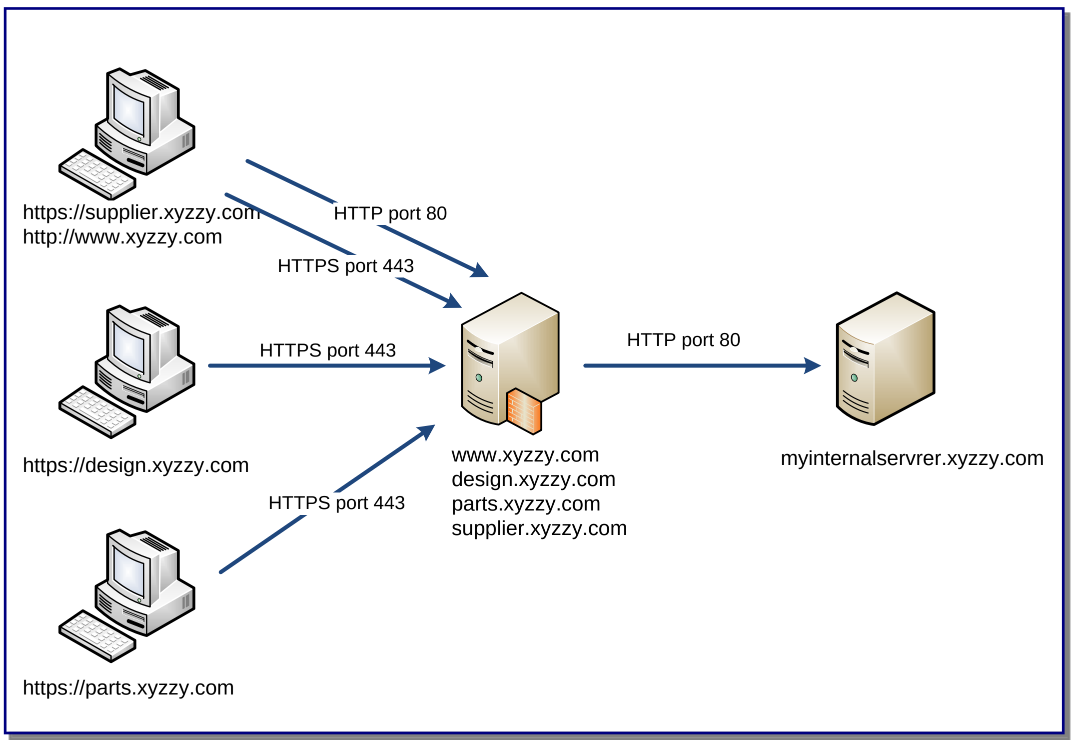


Figure 3-19, Real World Proxy Configuration

The fourth scenario (*Real World Proxy Configuration*) is a common configuration found on large sites. It is in fact just a compound on the previous advanced and protocol switching scenarios and, in terms of its configuration has the requirements of both.

3.9.2.2 HTTP Header

When a HTTP request is send to the server, the full URL is written in the browser, including the protocol, the host name, port, and path. However, within the HTTP protocol a slightly different format is sent. Let us assume that the following URL is entered in the browser: `http://server.domain.ext:1080/URL`. Then the browser will issue the following HTTP request to the server:

```
GET /URL HTTP/1.1
  Accept: */*
  Accept-Encoding: gzip, deflate
  Accept-Language: de,en-us;q=0.5
  Host: server.domain.ext:1080
  User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)
```

The request line (first line) contains the HTTP verb (`GET`), the URL to fetch (`/URL`) and the protocol version (`HTTP/1.1`). A 'Host: header' line is set containing the full name of the server that the browser sees as its communication partner. If the URL points to a non-standard port (HTTP uses by default port 80 and HTTPS port 443), then the port will be included in the Host: header.

The Host: header reflects the name that the browser uses to reach the server. It does not have to be the actual name of the server. It is just a name that can be mapped to an IP address to find the server, or via a proxy be resolved to the server. It is even possible that different names can be used to address the same server. The Host: header is the name (and port) that the browser thinks it is talking to (the server), and which will always be used for all subsequent HTTP requests.

For a small example, let us assume that on one Web Application Server (WAS), two different sites are actually served from the same instance of TC. Then one Reverse proxy server will have two names in the Internet, say `www.xyzzy.com` and `supplier.xyzzy.com`. The Reverse Proxy forwards all HTTP requests to the same server. The server must now use the name under which it was addressed, stored in the Host: header, (that has nothing to do with its own name!) for generating URLs and setting domain based cookies.

Many Teamcenter clients use the Host: header to generate URL's. For maximum flexibility you must always make sure the Reverse Proxy preserves the Host: header. That may not be always possible, for example Apache version 1 proxy cannot preserve the Host: header. Only from Apache version 2 and later is it possible to configure Apache so that the Host: header is forwarded unchanged. For this, set the configuration option **ProxyPreserveHost**. The Host: header must be preserved by proxy. The proxy must not in any way change the Host: header.

If it is not possible to preserve the host then a preference can be used to force the host name to be used. For Teamcenter Thin Client this preference is **Web_force_proxy_host**.

3.9.2.3 Protocol Switching & Redirect Loops

The third scenario with a protocol switch from HTTPS to HTTP is the most common option. This provides secure data communications over the Internet to the Reverse Proxy and also places the processor intensive work of handling the encryption with the Proxy (or associated hardware), and not on the WAS itself. On the final leg between the Reverse Proxy and the WAS server, normal HTTP is used. Although this is the most common scenario, it is also the most complex.

The typical problem situation that we have is that of a logon application that wishes to switch into HTTPS mode at some stage to securely transport data. First a check is made on the WAS to see if the application is already using HTTPS. Here the answer is no, as the incoming HTTP requests on the Web Server is using HTTP. So the application generates a new HTTPS URL, and orders the browser to redirect there. The next incoming request from the browser is HTTPS to the Reverse proxy, and then HTTP to the WAS. The application checks to see if it is HTTPS, finds that HTTP is in use, and then regenerates once again a HTTPS protocol. The situation now reached in the browser is that it continues in the redirect loop. Each HTTPS request sent to the Reverse proxy is answered with a redirect back to the same HTTPS URL! Another more relevant scenario is when the protocol is included in a URL. The server is checked to see what protocol it is running, HTTP and the resultant URL is sent to the client and of course it fails as the connection is in fact HTTPS.

There does not appear to be any general configuration solution to this problem. The advice is to avoid specifying the protocol in the web pages (HTML code) and allow the browser to default to the server's protocol. So if the server is running https any request to that server would also be in https. However, once again, absolute URLs require that a protocol be declared.

3.9.2.4 Redirection

If the application uses redirection, Teamcenter Project for example, then you will need to make use of **proxypassreverse** functionality to adjust the URL in the Location header on HTTP redirect responses.

3.9.3 Apache Reverse Proxy Configuration Example:

Reference the General Apache documentation and specifically the documentation for the Apache Proxy Module.

This example is of a mapping for a Teamcenter Thin Client using HTTPS both internally and externally, i.e. no protocol switch.

Note that the use of the longer form of the FMS URL (e.g. `/tc/fms/526080325/`) permits the use of the same port for both file and metadata traffic. The external address is `https://redirect.net.`*`mysite`*`.com:443`, the internal web server and FSC are both on server "Redirect" on ports 8888 and 4444 respectively. An example follows:

```
Listen 443
#################################################################
###################################################added for proxy
LoadModule ssl_module modules/mod_ssl.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule status_module modules/mod_status.so
LoadModule userdir_module modules/mod_userdir.so

<VirtualHost *:443>
   ServerName REDIRECT.net.mysite.com
   ProxyRequests off
   ProxyPreserveHost On
   RewriteEngine on
   SSLEngine on
   SSLCertificateFile C:\SIEMENS\reverse_proxy\key123\MYCERT.crt
   SSLCertificateKeyFile
      C:\SIEMENS\reverse_proxy\key123\MYKEY_CLEAN.key
   SSLCACertificateFile C:\SIEMENS\reverse_proxy\key123\MYCERT.crt
   ProxyRemote http://redirect.net.mysite.com:443
      https://redirect:8888/tc/webclient
   ProxyPass /tc/webclient https://redirect:8888/tc/webclient
   ProxyPassReverse https://redirect:8888/tc/webclient
      https://redirect.net.mysite.com:443/tc/webclient
   SSLProxyEngine on
   ProxyPass /tc/fms/526080325/ http://redirect:4444/tc/fms/526080325/
   ProxyPassReverse http://redirect:4444/tc/fms/526080325/
      https://redirect.net.mysite.com:443/tc/fms/526080325/
   ProxyPass /tc/ https://redirect:8888/tc/
   ProxyPassReverse https://redirect:8888/tc/
      https://redirect.net.mysite.com:443/tc/
</VirtualHost>
```

3.9.3.1 Forward Proxy Servers

A forward proxy is usually, quite simply a gateway between end-users inside the corporate Intranet and the Internet. Besides forwarding requests it might also provide other functionalities, such as content caching and filtering. Most commonly a forward-proxy will be a gateway from a LAN to a WAN.

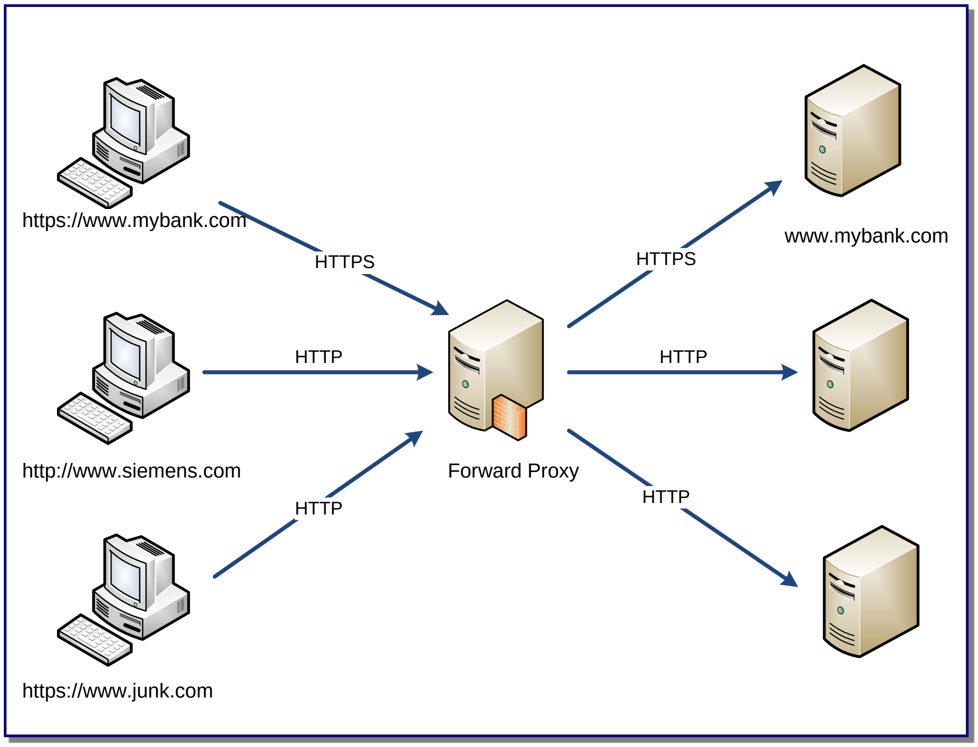


Figure 3-20, Forward Proxy

Here is an extract from an Apache configuration:

```
ProxyRequests On
ProxyVia On

<Proxy *>
Order deny,allow
Deny from all
Allow from internal.example.com
</Proxy>
```

3.9.4 *Proxy Authentication*

Proxy authentication is the login prompt you often receive when accessing a forward proxy server to access the Internet. It is not the same as server authentication such as is required by WebSEAL, Site Minder, etc. The term Proxy authentication comes from the HTTP 407 request and is not necessarily generated by the Forward proxy server. In many cases it is actually implemented within the corporate Firewall. The Client receives a 407 (proxy authentication required), which in the case of Basic authentication the browser prompts the user for username/password. Other forms of authentication are often used that do not require user interaction, for example NTLM (a Microsoft Authentication protocol) that uses the local username etc. to authenticate to the Proxy or Firewall.

Teamcenter only supports Basic authentication with the Rich client. NTLM support is planned for a future release.

Part of the 407 response is the ‘realm’. The realm serves two major functions. First, the client often presents this information to the user as part of the password dialog box. Second, it is used by the client to determine what password to send for a given authenticated area. The realm is displayed in the dialogue. Figure 3-21 shows an example for the realm called “Internet Access”.

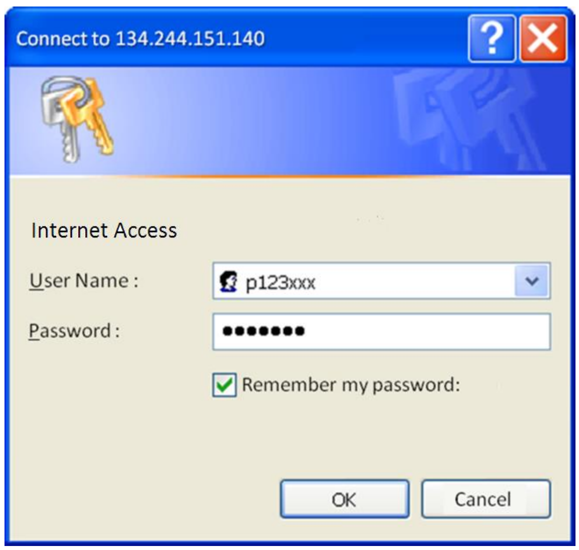


Figure 3-21, Typical Proxy Login Dialogue

It is possible to have several realms on the save server and support different authentication for each realm.

Here is an extract from an Apache configuration:

You will need a passwd file by using:

```
htpasswd -c /usr/local/apache/passwd/passwords userid
```

Next add to **http.conf**:

```
AuthType Basic
AuthName "Restricted Files"
# (Following line optional)
AuthBasicProvider file
AuthUserFile /usr/local/apache/passwd/passwords
Require user user
```

The ***AuthType*** directive selects that method that is used to authenticate the user. The most common method is Basic, and this is the method implemented by ***mod_auth_basic***. It is important to be aware however, that Basic authentication sends the password from the client to the server unencrypted. For highly sensitive data you should consider SSL i.e. ***mod_ssl***.

Note: **AuthName** is the realm.

3.9.5 Configuring Teamcenter for Proxy Servers

3.9.5.1 Teamcenter Thin Client

Provided the Reverse Proxy Server has been configured to preserve the HTTP Host, no client configuration is needed. Use the Reverse Proxy URL rather than the normal URL.

If it is not possible to preserve the Host then the reference **Web_force_proxy_host** must be used specify the node with which the URL's must be built. The node will be the external address of the proxy server. This setup will lose some flexibility.

FMS will use the proxy configuration of the browser, however more configuration is required, and this is described under the *FMS* section later in this document. The **FMSBootStrap_url** must include the proxy node in the list as this will be the route by which the Thin Client will access the FSC.

3.9.5.2 Rich Client

Two configuration changes are required to the client installation. First the server specified for connection in the **client_specfic.properties** file must point to the proxy server URL for the service.

Secondly the FCC.xml also has to point to the external URL for the fsc parent. Direct routing must be disabled to ensure the request goes to the FSC by the correct proxy URL.

3.9.5.3 FMS

The FSC used to serve clients on the intranet side of the Reverse proxy must have two entries in the **fmsmaster.xml** file; first the true entry for the intranet LAN, the second an alias entry with the external name and URL. The `clientmap` must be configured to assign this FSC to the external clients. The `parentfsc` or the FSC referenced by the `FMSBootStrap_url` will use its connection to the client to access the client's address; as the FSC is behind the proxy it will see the proxy servers address, not that of the client. Therefore you will need to use the proxy server's IP address when configuring the `clientmap` for this proxy FSC entry.

Reverse Proxy servers connected to the internet usually only allow traffic via Port 80 (or for https 443). One of the main features of a Reverse Proxy is the ability to redirect requests to different server based on the incoming URL.

For example `http://`*`tc.mysite/tc/controller`* requests can be sent to the Teamcenter Application Server. Ideally FMS requests should be sent to the same port. To support this FMS supports URLs in the following format `http://tc.mysite/tc/<`*`site id`*`>/fms` for example `http://tc.mysite/tc/12345/fms`. This allows the Reverse proxy to support more than one Teamcenter site by forwarding requests to the FMS server for each site based on the URL. This format of URL will be required for the `FMSBootStrap_url`, Parent FSC definition and in the proxy FSC description. An FMSmaster example follows.

In this example lines have been added for the alias entry for the FSC that is to be accessed through the firewall. In this case the true FSC (installed) is **FSC_REDIRECT_infodba** and the alias **REV_PROXY**. The `clientmap` IP address is the internal address of the reverse proxy.

```
<fmsworld>
  <fmsenterprise id="456360527">
   •
   •
   •
      <fscgroup id="mygroup">
<fsc id="FSC_REDIRECT_infodba" address="http://REDIRECT:4444">
   <volume id="1416000000081f5b5945"
      root="C:\\SIEMENS\\Teamcenter\\volume1" />
   <transientvolume id="7c98a09330fc22bcffd574494b847552"
      root="C:\\Temp\\transientVolume_infodba" />
</fsc>
<clientmap subnet="198.235.XXX.XX" mask="XXX.XXX.XXX.0">
   <assignedfsc fscid=" FSC_REDIRECT_infodba" priority="0" />
</clientmap>
      </fscgroup>
      <fscgroup id="rev_proxy_group">
<fsc id="REV_PROXY"
            address="https://redirect.net.mysite.com:443/tc/fms/
            526080325/">
</fsc>
<clientmap subnet="198.235.24.1" mask="0.0.0.0">
   <assignedfsc fscid="rev_proxy" priority="0" />
```

```
</clientmap>
      </fscgroup>
   </fmsenterprise>
</fmsworld>
```

3.9.6 Configuring Teamcenter for Forward Proxy Servers

3.9.6.1 Thin Client

In general the Thin Client inherits the Forward proxy settings of the Browser.

3.9.6.2 Rich Client

In the **FCC.xml** file the proxy server must be specified. Again both http and https can be supported. Example entries:

```
# These are required for proxy support:
#
#http.proxyHost=myproxy.mydomain.com
#http.proxyPort=8080
#http.nonProxyHosts=*.foo.com|localhost
#
# These are required for HTTPS proxy support:
# You can typically use the same proxy and port number
# for both HTTP and HTTPS proxying
# This may depend on your network topology, your use of
# proxying, and the capabilities of your proxy servers
#
#https.proxyHost=myproxy.mydomain.com
#https.proxyPort=8080
#https.nonProxyHosts=*.foo.com|localhost
```

3.9.7 Configuring Teamcenter for Protocol Switching

3.9.7.1 Thin Client

The Teamcenter Thin Client will not be able to automatically recognize protocol switching. The preference `WEB_protocol` must be used to tell the web application server to generate https based URLs.

3.9.7.2 Rich Client

Only the Rich client should be configured for https, Web application server will remain configured for only http.

3.9.7.3 FMS

Only the FCC should be configured for https. The receiving FSC(s) behind the server need only support http. For use with the Thin Client care must be taken to ensure the Thin Client is assigned to an FSC definition that specifies the correct node with the https protocol specified in the URL.

3.9.8 Configuring For Proxy Authentication

3.9.8.1 Thin Client

In general the Thin Client inherits the Forward proxy settings of the Browser. Anything supported by the browser will be supported by the Thin Client.

3.9.9 Rich Client and FMS

Rich Client and FMS use TcSS[32] to provide authentication support. This is documented in the *Security Services Installation/Customization* section of the *Teamcenter Help Library*. Only Basic Authentication and NTLM (Teamcenter 8.1) are supported.

[32] Teamcenter Secure Services

Chapter

4 Managing Enterprise Server Deployments

Go to main Table of Contents

Chapter

4 *Managing Enterprise Server Deployments*

This chapter describes deployment of Teamcenter Enterprise Servers, sometimes called the Business Logic tier, and presents sizing and tuning information.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- Server installation for: *Windows*, *UNIX/Linux*
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

4.1 Teamcenter Multi-tier Design

As Figure 3-1 in chapter 3 shows, one or more *Server Pools*, each coordinated by a *Server Pool Manager*, may be configured to serve the J2EE Web Application Server. For Rich Clients, each tcserver process in a pool contains state that ties it to a particular user, so that one tcserver cannot serve more than one client. Thin Clients keep no state information in the tcserver, however they do maintain affinity with a single tcserver process as long as it is available.

4.2 Enterprise Server Introduction

On the Enterprise Server, a *Server Pool Manager* (sometime referred to only as the "Server Manager") maintains a pool of business logic servers (tcserver processes). Each *Server Manager* maintains its respective pool, including the assignment state of each tcserver process, a number of pre-started 'warm' tcservers, how fast to start replacement tcservers, and how long to maintain tcservers no longer being used, among other things. There may be more than one such pool of tcservers on one or more machines.

The tcserver process provides the majority of Teamcenter's business logic and is relatively large and complex. This is the process that accepts user requests from the Web Application Server and communicates with the Database Server and File Management Servers (FMS) to satisfy those requests. Replies from the tcserver are returned to the client via the Web Application Server.

4.3 Server Manager Assigner Configuration

The Web Application Server places specific demands on tcserver assignments. This section documents some of the more applicable Server Manager parameters that can be tuned to optimize for this demand.

4.3.1 *Global Server Manager Configuration*

There are a few global *Server Pool Manager* parameters to consider that can be configured in the Web Application Server. This can be done within a '**globalPoolConfig.properties**' file placed in the J2EE Application Server startup directory, which the J2EE Application Server attempts to read at startup. If the file is not present, then default values are used. The .NET based pool server configuration parameters are stored in the %TC_ROOT%\net_servermanager\PoolConfiguration.xml file.

In the case where multiple web application servers connect into the same pool network, the first server instance to read a configuration value determines the value used by all other instances (in case of configuration conflict), hence the concept of "global" parameters. The default values generally perform well, but specific timeouts can be modified if need be. For more specific details, see the "*Installation on UNIX Servers*" or "*Installation on Windows Servers*" technical documents' Part IV "*Web Application Installation*". The table "Web Tier Optional Parameters" in that section has more information on the use of the following parameters; units for the following parameters are in seconds:

- `SOFT_TIMEOUT_EDIT`
- `SOFT_TIMEOUT_READ`
- `SOFT_TIMEOUT_STATELESS`
- `HARD_TIMEOUT_EDIT`
- `HARD_TIMEOUT_READ`
- `HARD_TIMEOUT_STATELESS`

4.3.2 *Local Server Manager Configuration*

Similarly there are server manager configuration parameters that can be tuned on a pool-by-pool basis. Their optimum values depend upon the specific characteristics of CPU power and available memory of the machine on which the Server Manager is running.

Thus, these parameters are considered "pool specific" parameters. They can be tuned via a configuration file (i.e. **serverPoolTEM.properties**) in the startup directory of each individual server manager.

- **PROCESS_MAX**: This determines the maximum number of tcserver processes the Server Manager will run concurrently in that pool. It should be configured based upon the amount of available RAM that can be devoted to the tcserver processes, divided by the average memory footprint of the tcserver processes. There may be situations where PROCESS_MAX should be adjusted downward from this value in the case of non-homogeneous machines configured in the same pool network. This is discussed in more detail later.

  NOTE

  If you intend to run the Server Pool Manager as a Windows service, see also SFB-Teamcenter-4555. The default value of Windows Desktop Heap for services may limit the number of tcserver processes to a number smaller than that specified by PROCESS_MAX. Some customers have reported an upper limit of fewer than 100 tcserver processes.

- **PROCESS_WARM**: This is the desired number of pre-started (i.e. 'warm') tcserver processes to have in reserve. Warm tcserver processes help ensure login requests over and above the target profile (see PROCESS_TARGET below) do not force users to wait for new ones to be started. In general this should be made higher if users complain about servers not being immediately available for login. A good or overly generous estimate for this value will reduce the importance of a good estimate for the PROCESS_TARGET demand profile below. Note that 'warm' tcservers do not consume licenses, only assigned tcserver processes do. The difference between PROCESS_MAX and PROCESS_WARM should not exceed the number of licenses available to run tcserver processes.

- **PROCESS_TARGET**: This is a series of comma-delimited time/value pairs that should roughly reflect concurrent session login demand over the course of a normal workday, e.g. "0700 10, 0730 50, 1200 10, 1300 80, 1900 10". Each time/value pair indicates the number of warm processes that should be available at that time. A good estimate here will reduce the importance of estimating a good value for 'PROCESS_WARM' (see above). Alternately, a generous flat profile (e.g. '0700 100') would probably be acceptable if the machine is dedicated to Teamcenter and sufficient physical RAM is installed.

- **PROCESS_CREATION_DELAY**: This parameter does not appear in the configuration file by default, but can be added in order to tune it. This is a series (vector) of increasing delays (in milliseconds) that the server manager waits between attempting to start each new tcserver process. The first value in the series is the default, normally 2000 (i.e. 2 seconds), e.g. "2000 2000 8000 16000 30000 60000". Normally only the first value is used. If the *Server Pool Manager* detects failures when starting tcserver processes, the subsequent longer values are used until startup failures cease.

If the machine has few and/or slow CPUs, then the first (and subsequent) value should probably be increased to prevent saturating CPU consumption with attempts to start new processes. Monitor CPU consumption during startup of tcserver processes, just after starting the *Server Pool Manager*.

To observe process startup CPU on a Windows system, open 'Task Manager' and select 'View' -> 'Select Columns…' and be sure that 'CPU Time' is selected. Then observe the cumulative CPU time for a tcserver process after all tcservers have entered the 'warm' state. Note that when all tcserver processes have been started to 'warm', the total CPU consumption on the machine should be near zero.

Similarly, for most UNIX platforms use 'ps' in a mode to show cumulative CPU consumption for the processes. Refer to the 'ps' man pages for your particular UNIX operating system.

For example, if it takes 8 seconds of CPU to start a new tcserver to 'warm' state and there are only two CPUs, then starting up tcserver processes every 4 seconds will consume all the CPU on the machine until target values are reached. However if there are 8 CPUs of the same power, then the system would only consume about half of the available CPUs starting one tcserver process every 2 seconds.

It is particularly important to tune this parameter in order to restore large backlogs in configured tcservers quickly, yet without overly saturating the system. Trying to launch many tcserver processes too fast for the CPU power <u>will result in high CPU load values and large run queues affecting all users. In extreme cases, it could result in startup timeouts where the server manager will think that tcserver startups failed.</u>

There are at least three situations where the server manager will have a backlog of tcserver processes to start, and the startup rate will be throttled by the PROCESS_CREATION_DELAY settings:

1. If the PROCESS_TARGET profile has a large step increase in the number of warm processes and the login demand is still below the next warm processes target in the series.
2. If the PROCESS_TARGET profile is flat for some period and higher than the session demand, and a burst of clients explicitly log-out. Explicit log-outs terminate the users' tcservers; then the server manager will start an equal number of replacement warm servers to restore the number in the profile.
3. If the PROCESS_TARGET profile is flat for some period and lower than the session demand, and a burst of clients log-ins consume a large number of 'warm' servers. Then the server manager will start replacement 'warm' servers to restore configured PROCESS_WARM margins.

4.4 Parallel Server Manager Pool Considerations

Configuring multiple Enterprise Server machines, that have multiple *Server Pool Managers* running within the same pool network, results in a somewhat more complicated PROCESS_MAX configuration. In such a configuration the J2EE Web Server *assigner*[33] will assign a tcserver from a pool that has fewer assigned tcservers relative to all other pool's PROCESS_MAX variable.

For example, if machine 'A' is running a Server Manager with PROCESS_MAX set to 100 and has 60 assigned tcservers, and machine 'B' is running a Server Manager with PROCESS_MAX set to 50 and has 25 assigned servers, then the assigner will assign a tcserver from the pool running on machine 'B'. In this example, machine 'B' had 50 percent of its configured maximum assigned, and machine 'A' had a higher percentage of 60 percent.

How can the ratios get out of balance? The answer is that the assigners have no control over when tcserver processes are dropped due to a client explicit logout or timeout. Servers are assigned to maintain the balance, but chance logouts will often unbalance the ratios. Notwithstanding, the chance of a very large imbalance is low.

There is no need for the administrator to consider this assignment algorithm in the case where each of the Enterprise Server machines (of the multiple pools) has the same CPU-power and memory. In such a case each Server Manager should be configured identically.

However if there is a significant mismatch in either CPU-power or memory, then there may be reason to carefully adjust the PROCESS_MAX lower on some machines than just available RAM might indicate.

Although it is not possible to predict how much CPU-load any individual tcserver assigned to a user will eventually place on the machine, over many assignments the tendency is that the CPU-demand will follow the *Law of Large Numbers*. With many tcservers assigned, the relative CPU-load will tend to be proportional to the number of assigned users; the more users assigned, the more accurate this tendency will be.

Let's consider an example consisting of three Enterprise Server machines with similar operating systems[34], but different CPU-power and RAM on each. Using the algorithm of setting PROCESS_MAX to the size of RAM divided by the memory demand per tcserver will tend to load balance the memory demand across the machines:

---

[33] A component of the Teamcenter J2EE Application loaded into the Web Server.

[34] so that memory-demand per **tcserver** process is identical

Table 4-1, Multiple Server Pool PROCESS_MAX Guidelines

Pool Machine	Machine[35] SPECint_rate2006	Machine RAM	CPU load balanced PROCESS_MAX	Memory load balanced PROCESS_MAX
A	150	4 Gig	130	105
B	200	8 Gig	175	210
C	250	8 Gig	220	210

However if the nature of the usage profile (see section 4.5.2) makes one or more of the machines CPU-limited rather than memory-limited, the relative PROCESS_MAX values can be reduced on some machines in order to balance CPU-load over to the machines with the higher CPU-power to memory ratios.

Adjusting the PROCESS_MAX value for a pool that is one of several in a pool network will also influence the considerations for setting the PROCESS_WARM values, since reducing it on any machine will tend to push more assignment load to the other machines. A large increase to PROCESS_MAX for a pool might generate a need to have its PROCESS_WARM value also increased.

Unfortunately estimating how much SPECint_rate2006 (SiR) capacity per assigned tcserver is consumed in every possible deployment is not realistic. The SiR is driven by the nature of the usage profile, which will vary significantly from business to business. The next section, *Enterprise Server Sizing Guidelines*, provides CPU and memory utilization as measured for a specific sample usage profile, which is detailed in Appendix A. But this serves only as a template to size *an initial pilot deployment*, from which the actual amount of CPU and memory consumed can be measured with your unique usage profile.

It is advisable for the administrator to keep historical records of system utilization, i.e. to occasionally employ system monitors and correlate CPU consumption with the number of assigned tcserver processes over the course of days or weeks. This should be done regularly if it is desired to document computing demand growth over time. This will also be valuable information for expansion planning or for specifying replacement hardware. See section 4.5.5, *Monitor System Usage*, for additional information to collect and track Enterprise Server utilization.

4.5 Enterprise Server Sizing Guidelines

This section provides guidelines for establishing initial server configuration and sizing requirements. It provides information about the types of computing resource required for

[35] SPECint_rate2006 machine ratings are a reasonable way to compare the relative computing power consumed by tcservers across non-homogeneous hardware over many transactions.

a specific sample usage profile[36], and aspects of each that may require adjustments to meet your unique usage requirements.

NOTE

Both the environments and usage profiles used to simulate Teamcenter operations, including scenarios, transactions, number and types of files, operating systems and databases have changed since the previous Teamcenter Deployment Guide was published. ***For these reasons it is not valid to compare estimates in this revision of the guide with previous revisions.*** Use the simulation information in this chapter and extrapolate from that to estimate the target pilot environment.

4.5.1 *Sizing Quick Reference*

This quick reference is to help locate Enterprise Server sizing information quickly for those who have already read this chapter. It assumes the reader has a good understanding of the factors that affect sizing information for both the Oracle and Enterprise servers and is familiar with terms defined later in the chapter (e.g. Oracle Demand Rate (ODR), Server Demand Rate (SDR), Usage Profiles, user categories, etc.).

The tables below provide guidelines for initial sizing estimates *of pilot deployments* only. Do not use this information until you have reviewed this section in its entirety.

NOTE

Starting with TC 11.2.0, usage profiles for Thin and Rich Client are significantly different. The resource requirements for Thin and Rich Client are therefore not compatible at any tier.

[36] as measured in the Teamcenter APA Scalability Lab

Table 4-2, Enterprise Server Sizing Information

Component	**Guideline**
CPU	Assuming the Usage Profile matches APA Thin and Rich Client Benchmarks;

Peak and average SDR per user for supported platforms is listed below as SPECint_rate2006 values for the measured usage profile:

	Thin Client		**Rich Client**	
Platform	**Peak SiR06 / User**	**Peak SiR06 / User**	**Avg SiR06 / User**	**Peak SiR06 / User**
AIX ORA	0.282	0.094	0.375	0.083
Solaris ORA	0.045	0.016	0.044	0.016
Suse ORA	0.072	0.034	0.049	0.021
Windows ORA	0.170	0.071	0.046	0.028
Windows SQL	0.135	0.070	0.045	0.023

Therefore, multiply the # of users by the average SDR and factor in an operating range reserve of 20% to handle the login rate.

For example, using 2000 Rich Client users on Suse:

Step 1) 0.049 x 2000 = 98

Step 2) 98 ÷ 80% = 122.5

Step 3) Select a Suse system with a SPECint_rate2006 rating of 123 or more.

Table 4-2, Enterprise Server Sizing Information

Component **Guideline**

Memory

Each *Concurrent* user consumes approximately the amount of RAM and SWAP listed below in Megabytes for the measured usage profile.

	Thin Client		Rich Client	
Platform	**MB RAM / User**	**MB SWAP / User**	**MB RAM / User**	**MB SWAP / User**
AIX ORA[37]	195.9	1.3	100.1	1.1
Solaris ORA	154.2	166.9	141.6	154.1
Suse ORA	87.3	-	91.4	-
Windows ORA	77.3	-	89.4	-
Windows SQL	69.7	-	84.7	-

Note that Solaris pre-allocates SWAP space when a process is instantiated, even though no actual paging may occur.

4.5.2 *APA Benchmark Usage Profile*

Customers routinely request sizing recommendations based purely on an estimate of the number of users they expect to access the system. Unfortunately, it is simply not possible to make an accurate recommendation based on user numbers alone. *Which product features are used, and how often, is the predominant system sizing consideration.*

The foundation of an accurate estimate of system resources for Teamcenter therefore begins with a full understanding of how the system will be used. This is referred to as a *Usage Profile*. Each profile consists of various elements such as the number and types of users, features accessed, login rate and frequency, to name a few. Key elements of a usage profile include:

- Number of users
- Types and categories of users
- Named vs. concurrent vs. active user ratios
- User login rate
- Features used by each user type
- How frequently users use features

[37] This is the actual RAM used with AME factor set to 2.0 (see section 4.5.3.2). Memory measured by SAR and other utilities will be twice this value.

The sizing information provided in this section is based on a sample Teamcenter *Usage Profile*[38]. It is referred to as the APA Benchmark Usage Profile. This profile may or may not be similar to yours. Although the APA Benchmark Usage Profile is not implemented identically in Thin and Rich Clients, the two are similar enough to allow comparison.

NOTE

You should obtain empirical system utilization data specific to your anticipated usage from your pilot deployment, and then extrapolate that to estimate the system resources needed for the production environment. A pilot implementation is highly recommended to obtain this empirical utilization data before sizing for production.

4.5.2.1 User Login Rate

One significant component of any *Usage Profile* is the "User Login Rate". This has the potential to affect Web application server CPU utilization more than anything else. User Login Rate is defined as the average number of login attempts per minute during any given period. If the User Login Rate is high then system resources will need to be increased to handle the peak load of users logging in plus users already logged in.

Looking at the chart in Figure 4-1 login rate is illustrated for 10,000 users. It shows it has an average rate is 88 logins per minute.

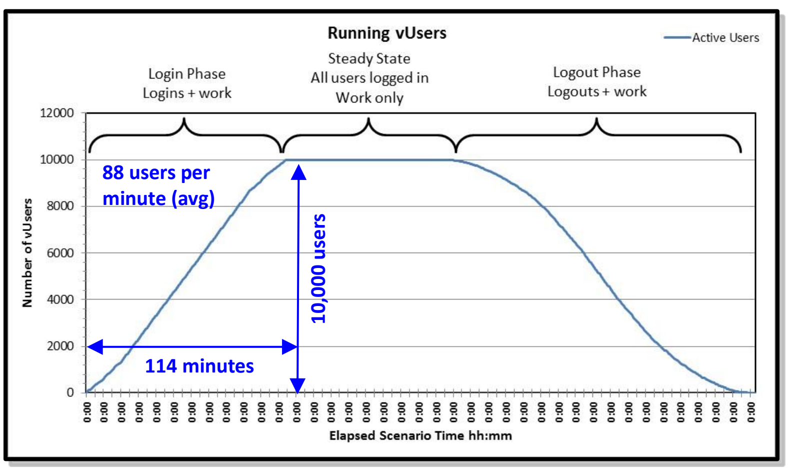


Figure 4-1, Example APA Benchmark User Login Profile

4.5.2.2 User Types

Users typically rely on a subset of Teamcenter features according to their role in the organization. Some users may work exclusively in Structure Manager or Requirements Manager, while others may work mostly NX via the Teamcenter Integration for NX feature. Generally most Teamcenter Thin Client users reference or retrieve information

[38] refer to appendix A, *Sample User Scenarios*, and appendix B, *Sample Usage Profiles*, for details

(i.e. consumers) and a small number create or manage that information (i.e. authors), while Rich Client users execute a more balanced mix of author and consumer activities. Take some time to document the types of users in your Teamcenter deployment. This will help determine usage profiles more accurately. Table 4-3 lists some sample user types:

Table 4-3, Sample User Types

Category	Description	Features Typically Used
Author	Creates documents, CAD files, items…	May use all Teamcenter features
Reviewer/ Approver	Reviews and/or approves information created by authors	Searching, Visualization, SM, Workflow, Referencers…
Consumer	Retrieves documents and CAD files, or simply reviews item information from master forms	Searching, Visualization, SM...
Configuration Management	Manages product assembly structure and configurations	Structure / Change Management

4.5.2.3 User Categories

Every type of user will fall into one of the following user categories and each will utilize system resources differently:

Table 4-4, User Categories

Category	Description	Resources Typically Consumed
Named User	User who has an account	- Database and Volume server storage
Connected	User who is actually logged in	- a concurrent license - Enterprise, and Oracle server ram & swap - Minimal CPU except for that required to complete the login (see below)
Active	User who is actually using the system	- Web, Enterprise, Oracle, FMS, & Flex server ram & swap - Additional CPU, ram, & swap depending upon the type of features being used (see below)

Table 4-4, User Categories (cont.)

Category	Description	Resources Typically Consumed
Concurrent	Users who are processing at the same time as others	Multiples of Active

As Table 4-4 shows there are other factors to be considered besides the number of users when sizing the Teamcenter Web environment. A system with thousands of *Connected* but only a few *Active* users may require fewer CPU resources than one with a hundred full time, active *Concurrent* users.

4.5.2.4 Features Used

Each feature provided by Teamcenter requires differing amounts of each type of computing resource. Your usage profile will likely employ a different set of features, and how often those features are used. For example, one feature may require pieces of data from different sources such as a .gif image from the volume plus a simple query of the database, thus requiring more work from the Enterprise and FMS servers, but very little from Oracle. Another feature may not require any volume data but requires a complex query to be performed on the database placing a heavy load on Oracle but little on the Enterprise server and virtually none on the FMS server.

The best gauge to estimate system resource requirements is empirical measurement of a pilot implementation that approximates the expected usage profile of the target production environment. Resource utilization of the pilot system can then be extrapolated to production usage volumes.

To size platform requirements accurately for Teamcenter Enterprise Servers, it is helpful to classify users into groups, estimate the number of users in each group, and then assign the types and number of activities each group of user is expected to execute.

4.5.2.5 Usage Profile Accuracy

Unfortunately, it can be very difficult to estimate accurate usage profiles until some actual operational experience is acquired. Experience shows that once users have access to a new system, the methods they use to accomplish tasks differ from the methods they used before. As a result, it is not uncommon for actual usage to be different from that originally anticipated. A pilot implementation will help validate the expected production *usage profile*.

4.5.2.6 Demand Rate

A *Usage Profile* will generate a specific system *Load Profile* that characterizes what resources are used in each server throughout a typical day. Think of load profile as the pattern of daily system utilization. That is, the *Demand* users place on the various Teamcenter servers, or, how much work the servers must perform. The load profile can be simplified by averaging daily utilization and adjusting for peak usage to estimate a *Demand Rate*.

In this document, *Demand Rate* is denoted as a *SPEC* rating, a popular indicator of how much work a computing system can perform (i.e. throughput). For multi CPU servers the *SPECint_rate* rating gives a good indication of the system's throughput capacity. *SPECint_rate2006*, the current benchmark used by platform providers to rate server throughput, is used throughout this document to estimate CPU capacity. Some of the hardware used as references in this document have only *SPECint_rate2000* ratings, SPEC's obsolete throughput benchmark. As a result, some CPU recommendations specified as a *SPECint_rate2006* value are derived from *SPECint_rate2000* values, or from benchmarks of similar systems.

SPECint_rate2006 ratings for many commercial computing systems are available on the web from the Standard Performance Evaluation Corporation (SPEC) at:

> http://www.spec.org/cpu2006/results/

Information about SPEC and other benchmarks are available at:

> http://www.spec.org/

4.5.3 *Enterprise Server Sizing*

Teamcenter Enterprise server size varies depending upon the anticipated *Usage Profile* (refer section 4.5.2), client type, and server hardware platform.

Teamcenter Enterprise servers consist of at least one *Server Pool Manager* process and dozens to thousands of tcserver processes. For example, when calculating the memory requirements for Enterprise Server, include resources for:

- The *Server Pool Manager* CPU, memory, and swap
- Total CPU, memory, and swap for all Teamcenter server processes (n * tcserver)

Measurements reveal that the resource requirements of the *Server Pool Manager* process are very small regardless of *Usage Profile*. But be sure to also include any other running processes such as those needed by the operating system.

Table 4-5 illustrates per user resource consumption of the Enterprise Tier server hosted on various hardware / OS combinations, as measured in the Teamcenter APA performance and Scalability Lab with the standard Rich Client APA usage profile (see Figure 4-1). For comparative purposes, CPU utilization is represented in SPECint_rate_2006 (SiR) values and rounded up to one thousandth, per user.

Overall, average CPU requirements for the Teamcenter 11.2.1 Enterprise tier have increased noticeably depending on platform, AIX, Suse, and Windows swap / pagefile requirements are minimal (deltas don't apply).

NOTE:

For AIX memory measurements, AIX Active Memory Expansion (AME) was enabled. AME reduces the memory resource required by the ratio specified as an 'Expansion Factor' parameter when configuring the machine. For Tc 11.2.1 measurements, an expansion factor of 2.0x was configured, effectively doubling available RAM.

Table 4-5, Enterprise Server CPU, Memory, and Swap Usage – Rich Client

Teamcenter Version	Platform	Number of Users measured	Peak SiR06 / User	Avg SiR06 / User	MB RAM / User[39]	MB SWAP / User
Tc 10.1	AIX ORA	500	0.260	0.065	68.5	1.1
	Solaris ORA	2000	0.023	0.009	123.7	141.0
	Suse ORA	1500	0.040	0.017	106.1	-
	Windows ORA	2000	0.033	0.014	67.5	-
	Windows SQL	2000	0.028	0.015	61.5	-
Tc 11.2.1	AIX ORA	500	0.375	0.083	100.1	1.1
	Solaris ORA	2000	0.044	0.016	141.6	154.1
	Suse ORA	1500	0.049	0.021	91.4	-
	Windows ORA	3000	0.046	0.028	89.4	-
	Windows SQL	3000	0.045	0.023	84.7	-
Deltas	AIX ORA		44.4%	26.8%	46.2%	0.2%
	Solaris		88.9%	75.5%	14.5%	9.4%
	Suse		23.1%	21.8%	-13.9%	
	Windows ORA	1000	38.4%	95.4%	32.4%	
	Windows SQL	1000	57.6%	54.1%	37.6%	

4.5.3.1 Enterprise Server Memory Sizing

Figure 4-3 illustrates memory consumption of the server pool for the APA standard Rich Client usage profile, on a Sun SPARC-T7-2 @ 4133 MHz (LDOM).

[39] For AIX, this is the actual RAM used with AME factor set to 2.0 (see section 4.5.3.2). Memory measured by SAR and other utilities will be twice this value.

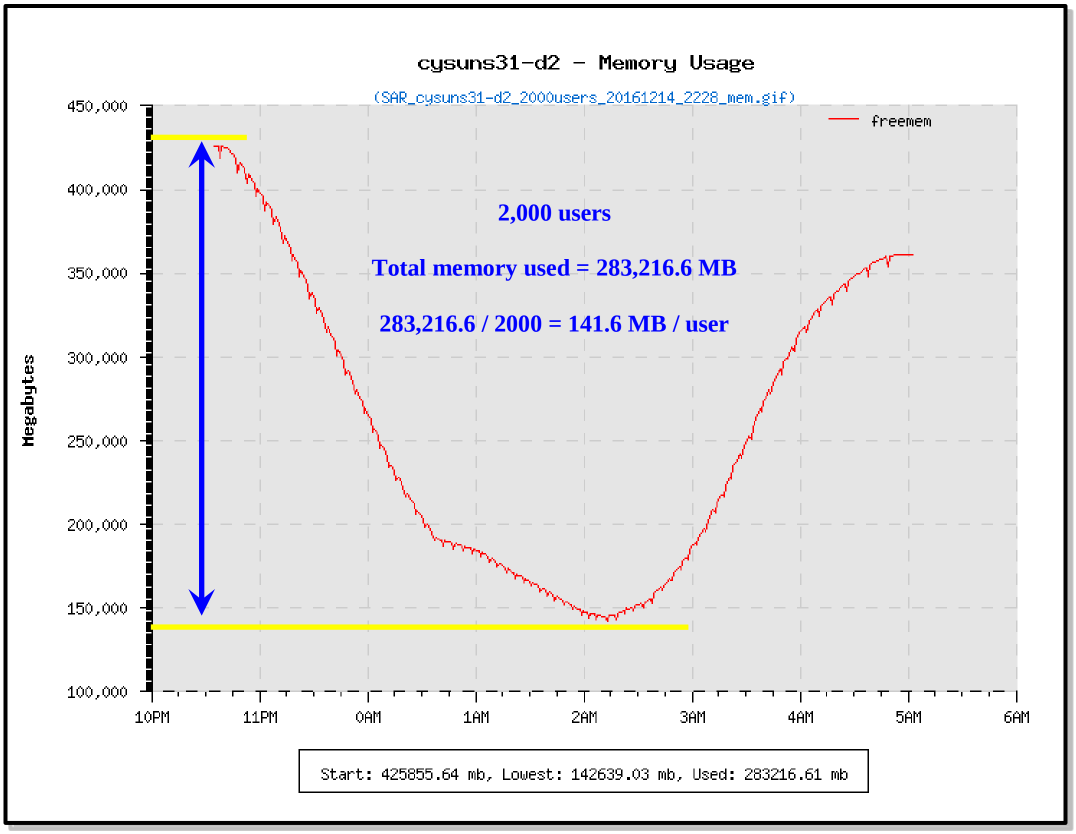


Figure 4-2, Rich Client Enterprise Server RAM Utilization – Solaris

Dividing the total memory used by number of tcserver processes yields memory required per process. This profile results in a tcserver process for each of the 1000 logged in users. With this profile, each Teamcenter tcserver process consumes approximately 142 MB of physical ram on Solaris, or 126 MB per *logged in* user. Your *Usage Profile* may be significantly different.

4.5.3.2 Enterprise Server Memory Sizing on IBM Power7+

Figure 4-3 shows an example of memory consumption of the server pool on a single IBM Flex System p260 server (7895-23X) with 108 GB physical ram. The IBM Flex systems support Active Memory Expansion (AME), which transparently compresses/expands memory below the Operating System layer. The amount of compression is set within each partition with an AME expansion factor value between 1 and 10. An expansion factor of 2.0 effectively doubles the available memory, thereby reducing the actual memory requirement for executing processes in half.

NOTE

Only the IBM Power 7 or later platforms currently support the memory expansion feature; there are no other known platforms that provide such capabilities.

As Figure 4-3 shows, with AME set to 2.0, that SAR reports the effective memory of 216 GB, and total used as 195.8 GB. Actual memory used for this profile then is 97.9 GB.

Dividing the total memory used by number of tcserver processes yields memory required per process. This profile results in a tcserver process for each of the 1000 logged in users. With this profile, each Teamcenter tcserver process consumes approximately 98 MB of physical ram on AIX, or 98 MB per *concurrent* user. Again, your *Usage Profile* may be significantly different.

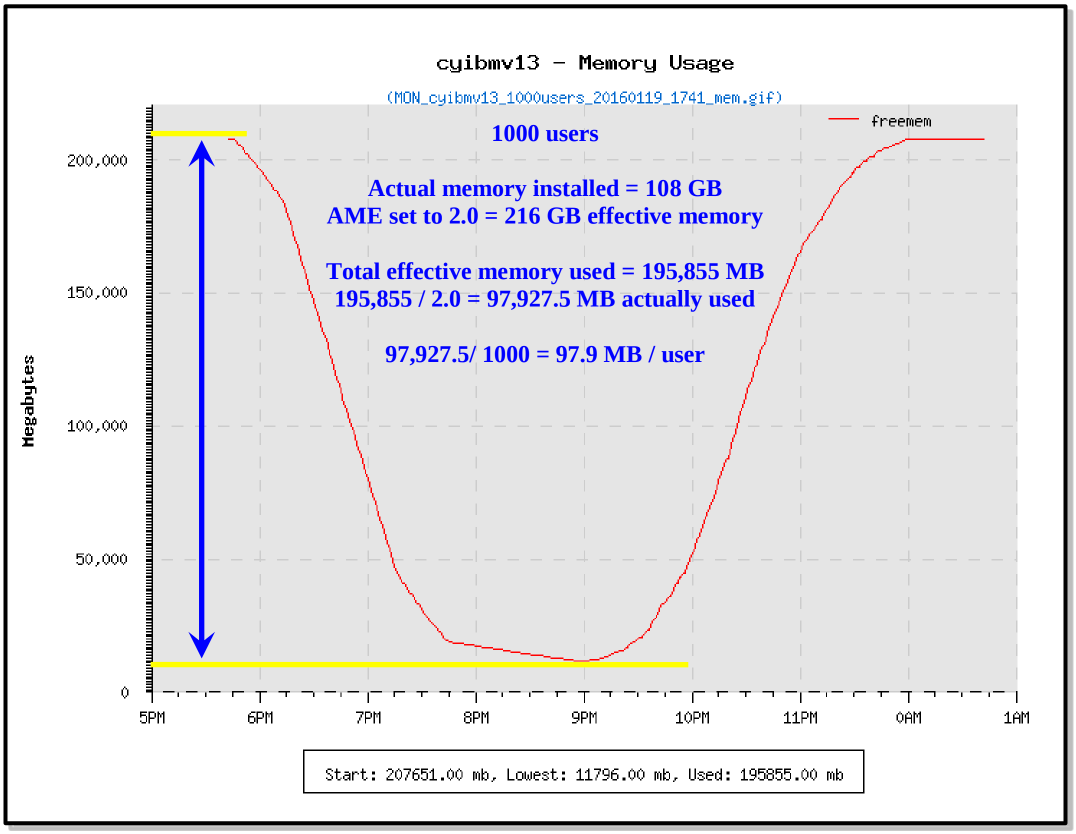
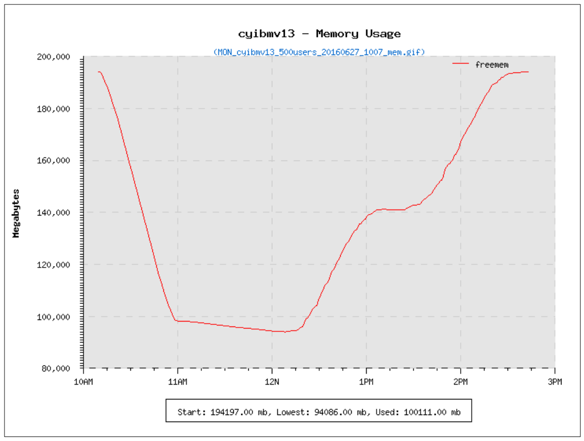


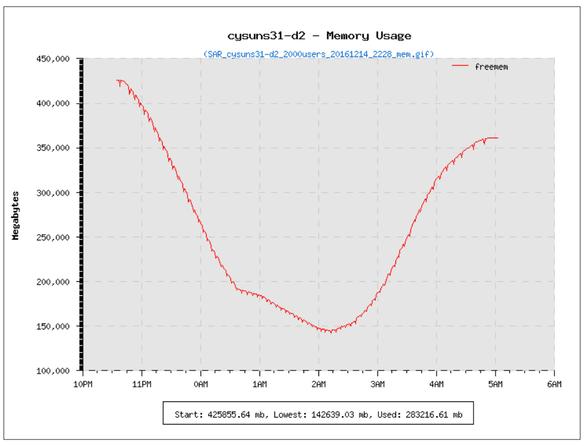
Figure 4-3, Rich Client Enterprise Server RAM Utilization – AIX (AME = 2.0)

As shown in Figure 4-4, with the APA standard Rich Client usage profile, enterprise tier memory consumption profile is fairly similar across all platforms. These charts track available memory as users login, work, and then logout (freemem on UNIX). As more users log in, more RAM is consumed. Used memory is equal to the starting amount before the benchmark starts, less the lowest amount observed (Used = Start - Lowest).

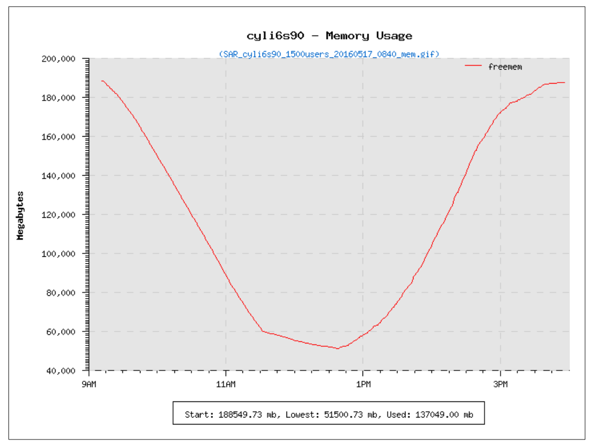
AIX Oracle – 500 users[40]



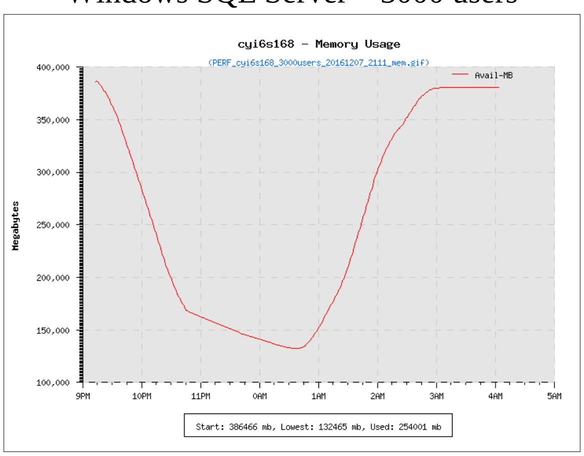
Solaris – 2000 users



Suse – 1500 users



Windows SQL Server – 3000 users



Windows Oracle – 3000 users

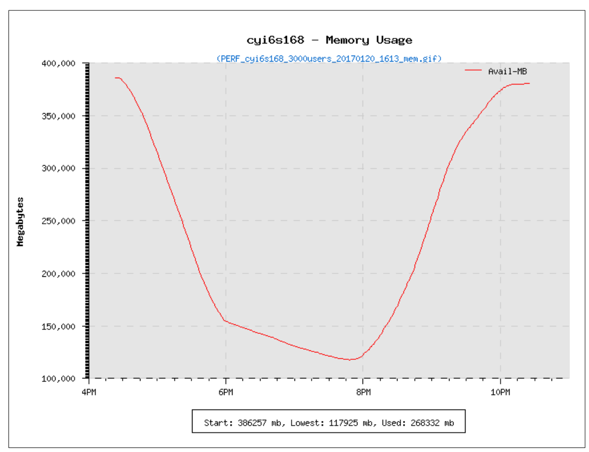


Figure 4-4, Rich Client Enterprise Server RAM Utilization – All Platforms

[40] This is the value of RAM measured with SAR. The actual RAM used with AME factor set to 2.0 will be half this value (see section 4.5.3.2).

Enterprise tier memory usage increased from the baseline release to various degrees on all platforms as seen in Figure 4-5. The red columns represent RAM, the blue columns SWAP/page. SWAP/page file use was at or near zero on all platforms except Solaris.

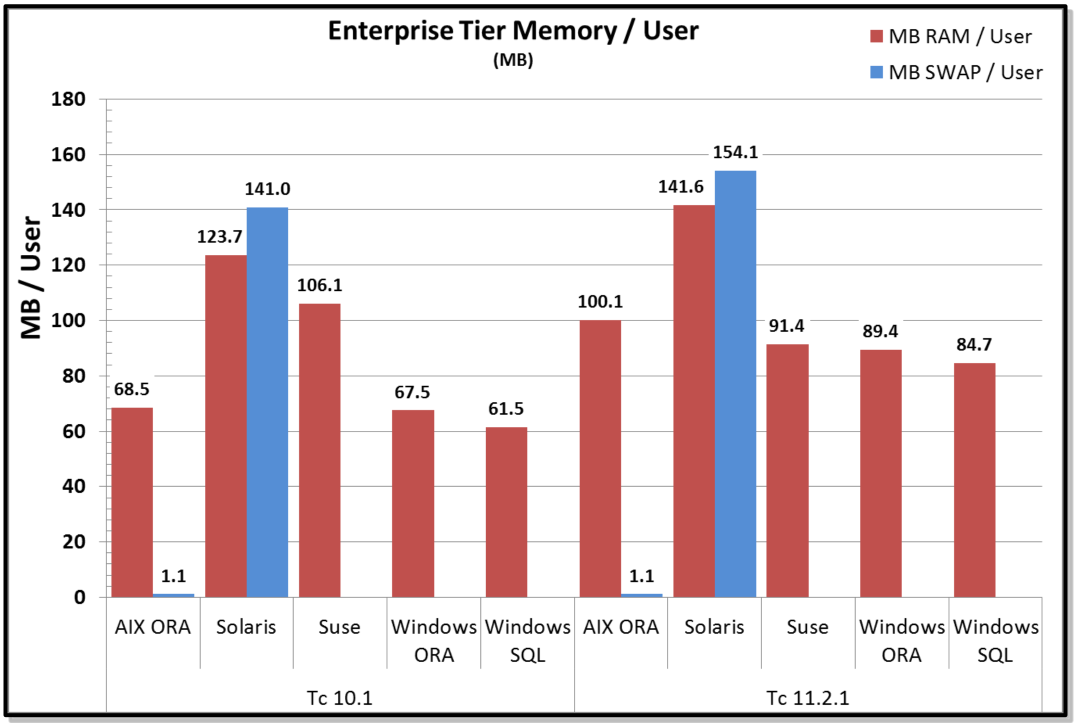


Figure 4-5, Rich Client Enterprise Server Memory Utilization per User

Note too that Teamcenter makes extensive use of shared libraries. As the number of tcserver processes decreases, shared libraries become a larger portion of the overall memory used.

Based on the measurements above, Table 4-6 provides an estimate of how many users an application server can support for the given amount of physical memory installed, with usage profiles similar to the APA standard Rich Client usage profile. These estimates assume adequate CPU capacity to support these numbers of users. The **Estimated # of Users** columns represent an approximate number of users the system can support before paging would occur. Above that number the system will likely continue to function but with increased end-user response times due to paging.

The projections in Table 4-6 are based on the APA standard Rich Client usage profile and assume each OS requires approximately 2 GB ram.

Table 4-6, Enterprise Server Memory Estimates

		AIX_ORA	Sol	Suse	Win_ORA	Win_SQL
RAM per User		**100.1 MB**	**141.6 MB**	**91.4 MB**	**89.4 MB**	**84.7 MB**
Physical Memory (GB)	**Usable Memory (GB)**	**Estimated Number of Users**				
4	2	10	10	20	20	20
8	6	30	40	60	60	70
12	10	50	70	110	110	120
16	14	70	100	150	160	160
24	22	110	150	240	250	260
32	30	150	210	330	340	360
64	62	310	440	690	700	740
128	126	640	910	1410	1440	1520
256	254	1290	1830	2840	2900	3070
512	510	2600	3680	5710	5830	6160
1024	1022	5220	7390	11450	11700	12360

4.5.3.3 Enterprise Server Swapfile/Pagefile Sizing

NOTE

Nearly all contemporary operating systems use virtual memory paging to accommodate programs that require more memory than is physically available, although some still do swap out entire programs under extreme memory pressure. However the term '*swap*' is typically still used with UNIX based operating systems even though memory is being paged.

For the Solaris operating system, enough space to completely swap out the program and its data is pre-allocated in the '*swapfile(s)*' when the program is started. If sufficient free space is not available in the swapfile(s), the program will not start. On Solaris therefore, it is important to be able to estimate the total swapfile space required for Teamcenter Enterprise Server deployments.

AIX, Suse, and Windows operating systems do not pre-allocate space in the swap/pagefile(s). For these systems it is not as important to estimate Enterprise Server swap/pagefile requirements. Figure 4-6 illustrates these differences.

Figure 4-6 shows swap / page usage for all measured platforms, illustrating how Solaris requires a significant amount of swap space, while AIX, Suse, and Windows require little or none. Note that UNIX charts show the amount of swap space available, while the Windows chart show percentage of the pagefile used. Also notice the Y-axis scale difference between various charts.

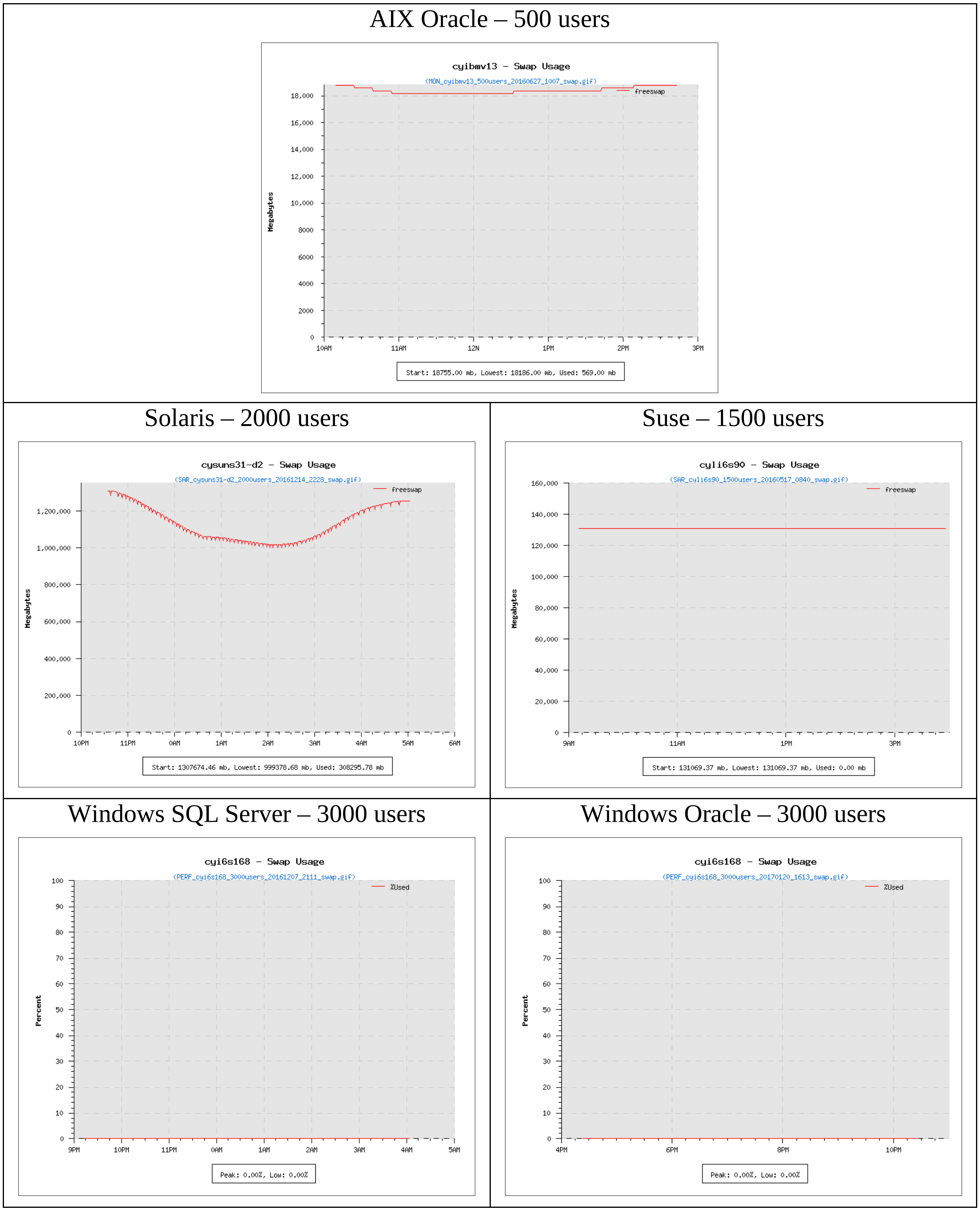


Figure 4-6, Rich Client Enterprise Server SWAP / Page Utilization – All Platforms

NOTE

To minimize paging and assure optimal performance, limit the number of users per server by the amount of physical memory installed in the machine (see section 4.5.3.1 above). The actual swap requirement per user may be different depending upon your production *Usage Profile*.

4.5.3.4 Enterprise Server CPU Sizing

If the Demand Rate can be determined (i.e. how many SPECint_rate2006s are needed) for the expected Teamcenter *Usage Profile* it can be cross-referenced against various systems listed by the SPEC organization to find one that is suitable. The value derived for the usage profile measured is referred to as the Teamcenter Server Demand Rate (SDR).

Determining the Enterprise Server Demand Rate (SDR)

As with all Teamcenter server tiers, the *Usage Profile* is the predominant factor in determining the SDR and ultimately the size of the Enterprise Server. For this server, a key sizing factor is the user login rate. Take some time to understand the login process and establish an accurate login rate in order to get an accurate SDR. Measurements that follow were obtained from scalability benchmarks using the APA Benchmark usage profiles had three basic user group login rates as follows:

- Early birds (small group) – arrive around 7:30am
- Normal crowd – around 8 to 9
- Stragglers (small group) – 9 o'clock on

(Reference Figure 4-1)

NOTE

Users typically don't just login and sit idle. These usage profiles assume users start working as soon as they login. This means they are putting an increasing load on the server; while these users are working other users are logging in so there are two concurrent loads to consider – 1) current usage, and 2) login.

CPU utilization on the Enterprise tier comprises the majority of CPU requirement in a Teamcenter deployment. For acceptable server response times, CPU utilization should be limited to a maximum of 80%.

Figure 4-7 shows CPU utilization for 3000 users using the APA standard Rich Client usage profile. The SPECint_rate2006 rating of the Dell server used for this benchmark is 723.0. At 9.66% CPU average and 18.6% peak, this equates to a Server Demand Rate (SDR) per user of 0.023 (723 x 0.186 ÷ 3000) at steady state (average) and 0.045 at peak (723 x 0.0966 ÷ 3000).

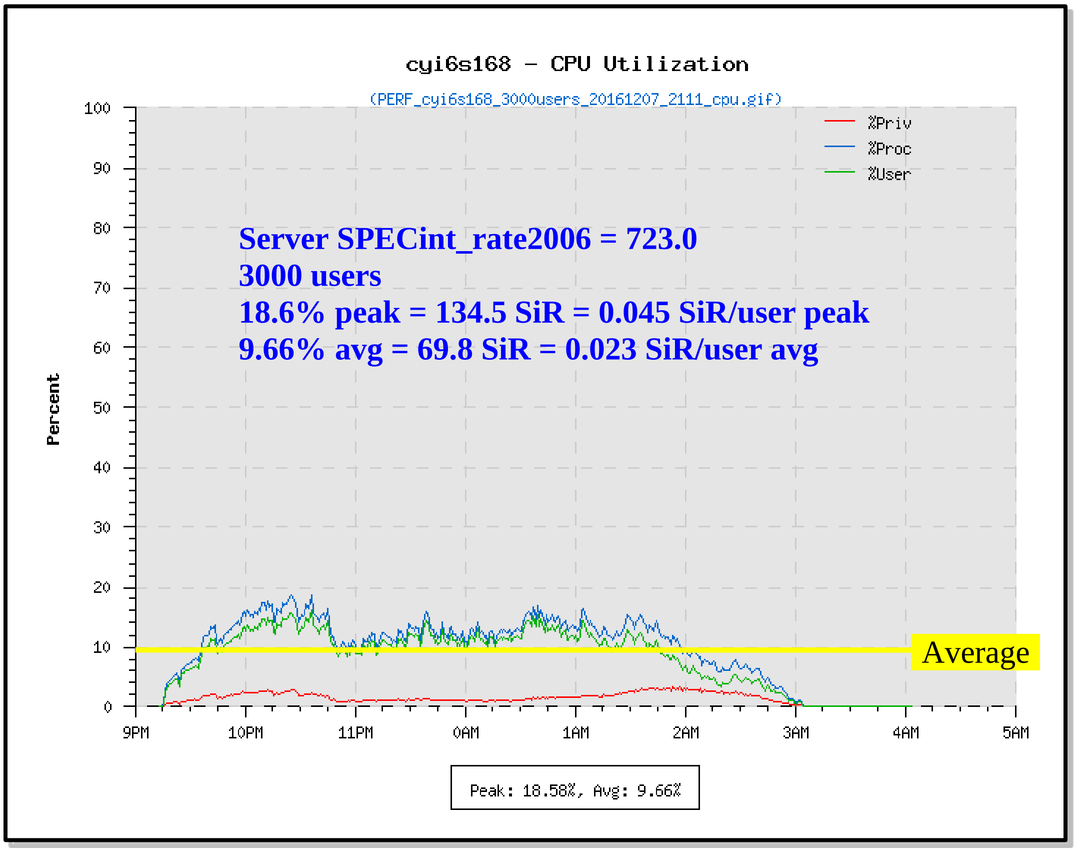
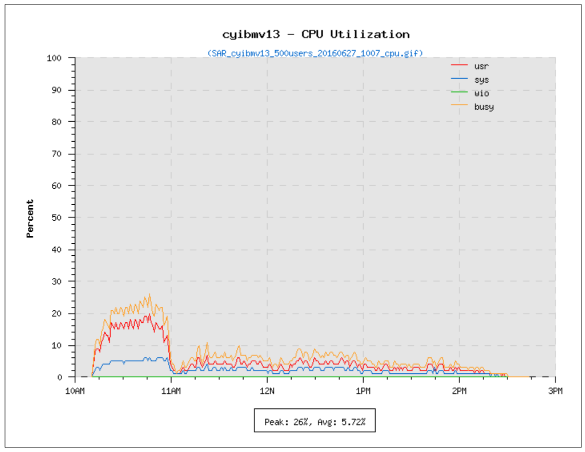


Figure 4-7, Rich Client Enterprise Server CPU Utilization – Linux

If the anticipated login period is short, or the login rate is expected to be less than that of APA Benchmark (40 users per minute), consider setting CPU requirements from the average, steady state SDR. In order to keep CPU utilization below 80% on an Dell x86 system running Windows with 1500 users, a system with a minimum SPECint_rate2006 value of 43.2 is needed to provide adequate performance for this profile (0.023 x 1500 ÷ 80%), or 84.4 if consistent performance is desired (0.045 x 1500 ÷ 80%).

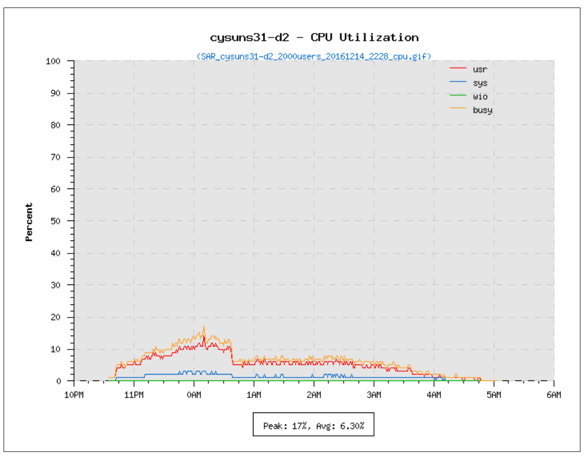
As the CPU charts show in Figure 4-8, utilization can be significantly different on various platforms. Note the increased CPU during the login period.

AIX Oracle – 500 users



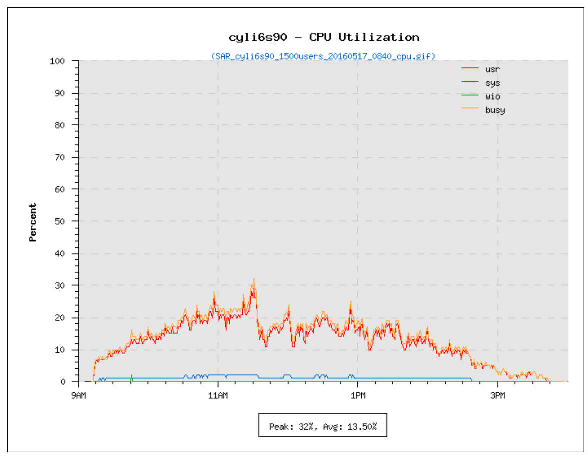
(14 x POWER 7+ @ 4116 MHz)

Solaris – 2000 users



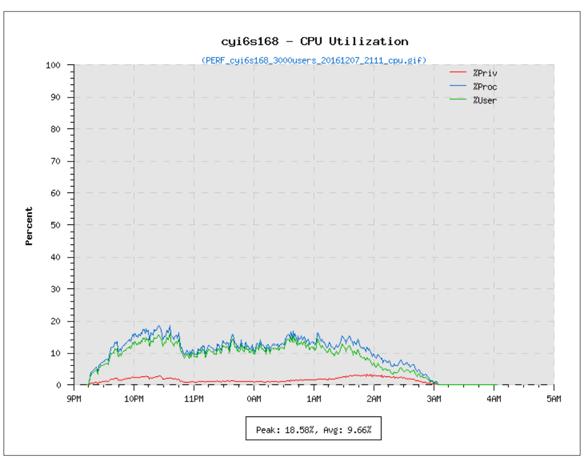
(12 x SPARC-T7 @ 4133 MHz)

Suse – 1500 users



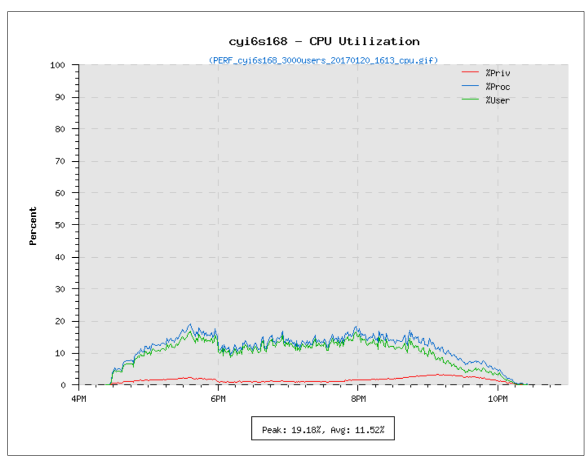
(8 x Intel® Xeon® CPU E5630 @ 2.53GHz)

Windows SQL Server – 3000 users



(16 x Intel® Xeon® CPU E5+2640 @ 2.6 GHz)

Windows Oracle – 3000 users



(16 x Intel® Xeon® CPU E5+2640 @ 2.6 GHz)

Figure 4-8, Rich Client Enterprise Server CPU Utilization – All Platforms

This can be more or less than the baseline release by some margin as seen in Figure 4-9. The blue portion of the columns indicates average CPU, the red portion peak CPU. Note that Enterprise tier average CPU increased on all platforms compared to the baseline version, especially AIX.

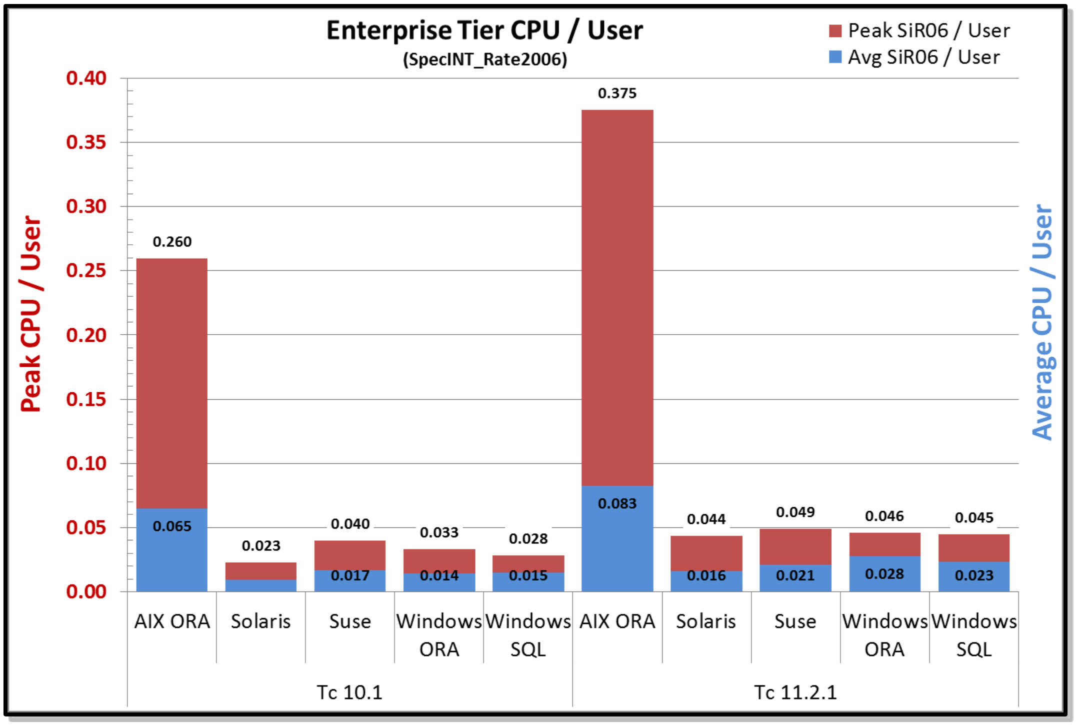


Figure 4-9, Rich Client Enterprise Server per User CPU Utilization vs. Baseline

4.5.4 *Impact of Overloading Enterprise Server CPU*

Overloading the CPU at the Enterprise Server tier will have a direct adverse impact on response times. Two CPU metrics are critical to determine if the CPU resource is overloaded;

1) % CPU utilization, and
2) CPU run queue[41].

Measurements have shown if % CPU exceeds 90 *and* run queue exceeds 1.5-2x the number of CPU cores[42], the CPU resource is likely overloaded.

To illustrate this, a series of experiments were conducted in the APA labs to purposely overload the Enterprise server and assess the effect on server response times. Since it is known that login creates a significant CPU load, the number of users logging in within a set period of time was incrementally increased until the CPU saturated at 100% during the login phase.

---

[41] Number of processes waiting for CPU resource at any particular time.

[42] Cores, not threads (or virtual CPUs)

Figure 4-10 shows the period of time when login rate is highest between the two red lines, which presents the heaviest load during the course of these benchmarks. Figure 4-11 shows the resulting CPU utilization.

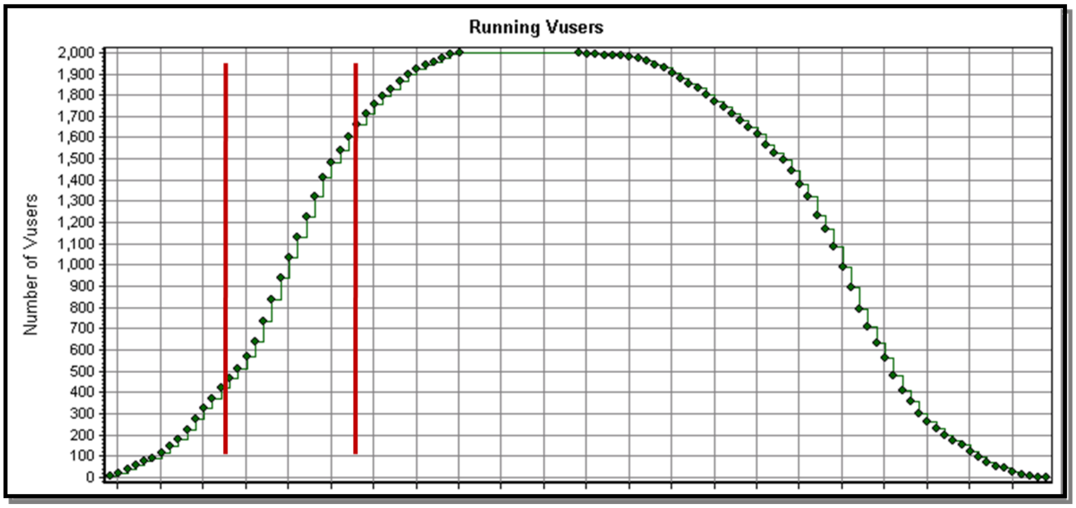


Figure 4-10, APA Standard Usage Login Profile

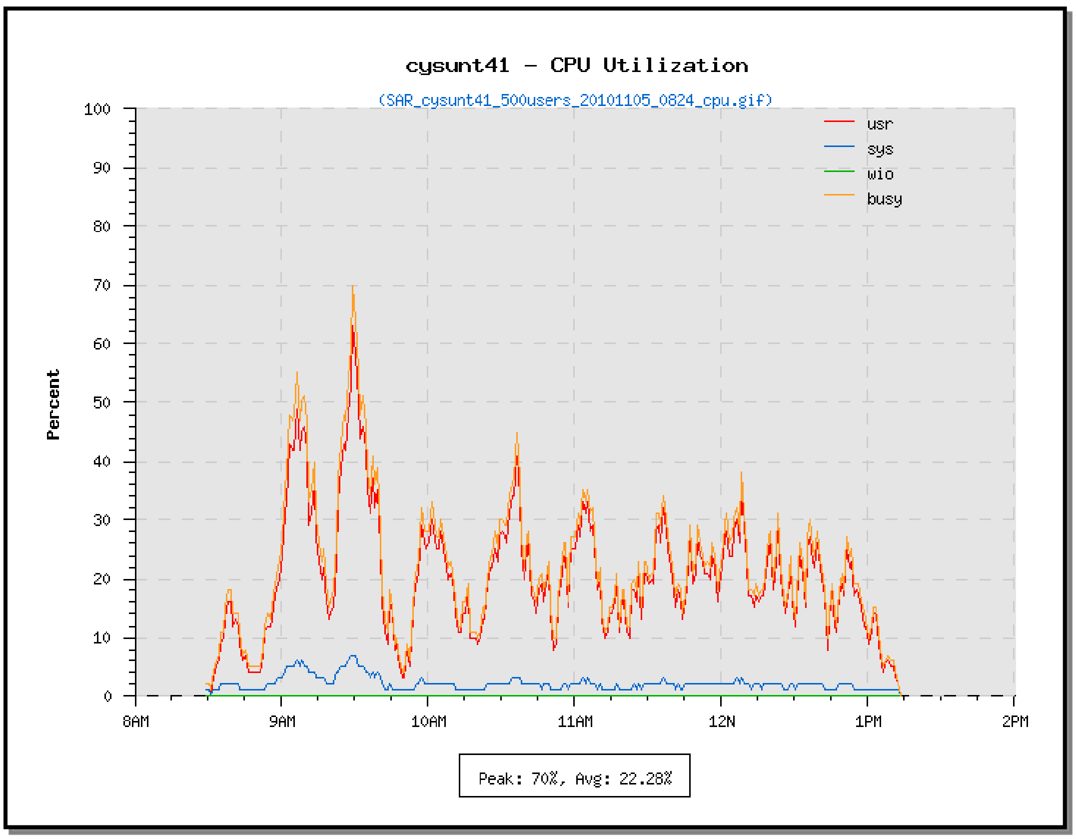


Figure 4-11, Sample CPU Profile – 500 Users

Increasing the number of users logging in during this timeframe directly increases CPU utilization during the login phase. Figure 4-12 shows CPU approaching 100% for a very brief period of time, with the user login profile overlaid in green.

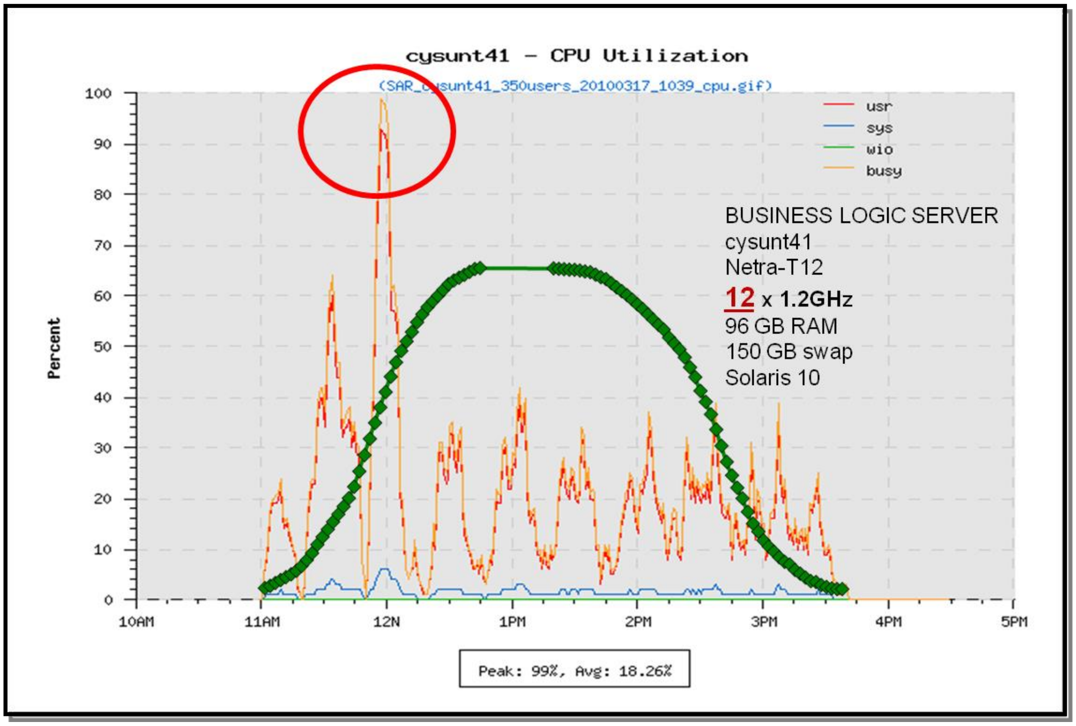


Figure 4-12, Approaching CPU Saturation

This results in a very high percentage of running processes and processes waiting for CPU. The blue curve in Figure 4-13 shows %runocc, which is the percentage of time that <u>at least one process</u> is waiting in the CPU run queue, that is, waiting for CPU. runq-sz, in red, shows <u>the number</u> of processes waiting for CPU. At 99% utilization %runocc peaks at 75% and a peak[43] of 4.2 processes are waiting for CPU as seen in Figure 4-13.

[43] Average of peak values during the measurement period

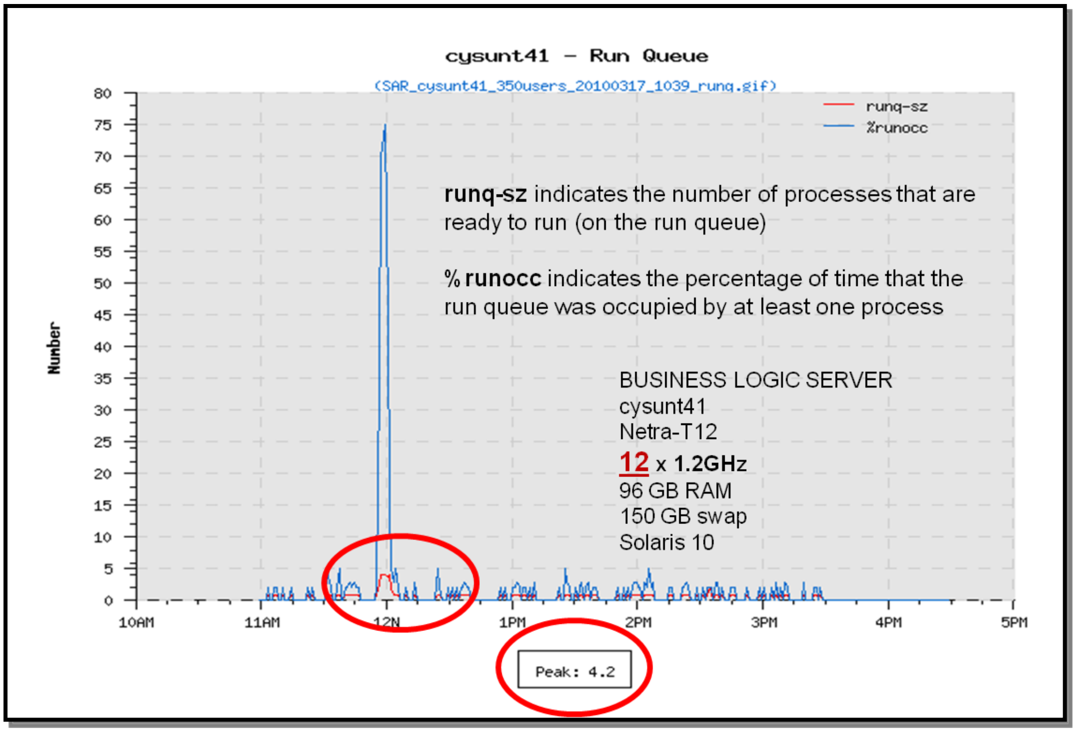


Figure 4-13, Acceptable CPU run queue size (runq-sz)

With 12 CPUs, the system is able to keep up, but is on the verge of becoming bottlenecked by CPU. Figure 4-14 shows server response times remain stable throughout the run.

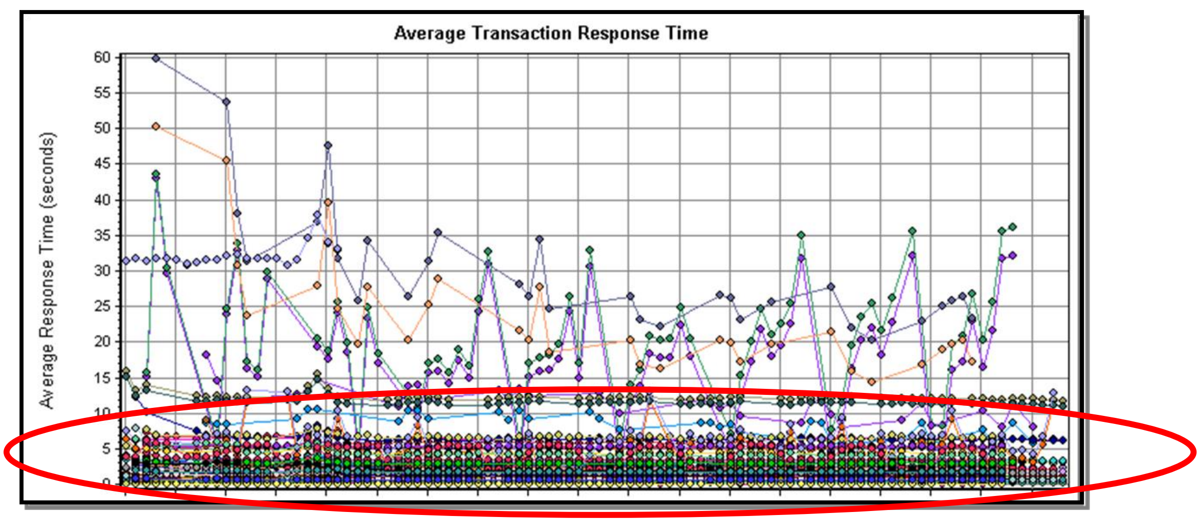


Figure 4-14, Acceptable Server Response Times

Further increasing login rate pushes CPU to 100% for several minutes (Figure 4-15).

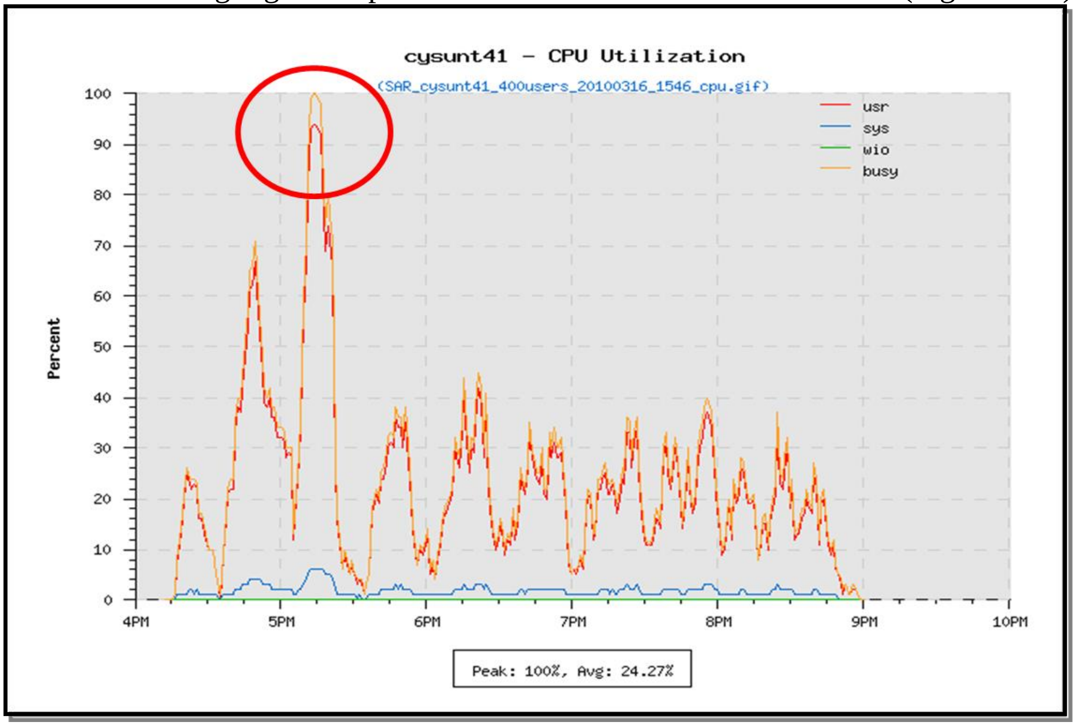


Figure 4-15, Momentary CPU Saturation

The result is 100% %runocc and even higher runq-sz (Figure 4-16). At 17.0, runq-sz is still below the recommended maximum value of 1.5-2 times the number of CPUs.

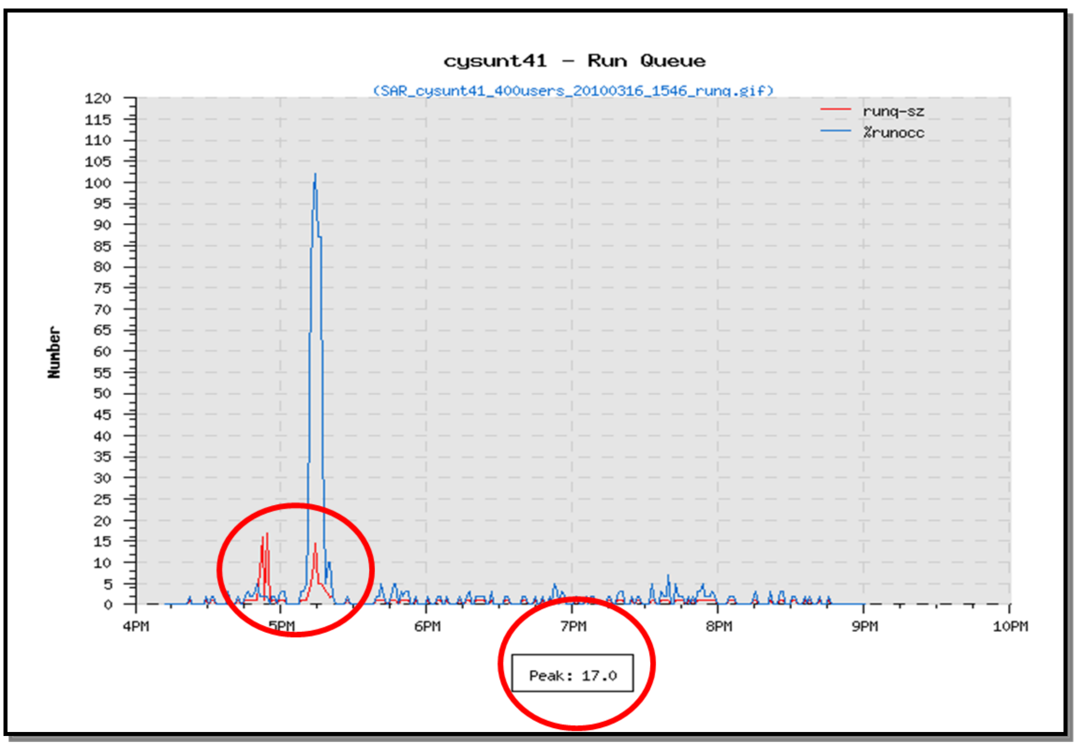


Figure 4-16, Increasing CPU Run Queue Size (runq-sz)

Server response time now begin to show the effect of CPU bottleneck during the login phase (Figure 4-17).

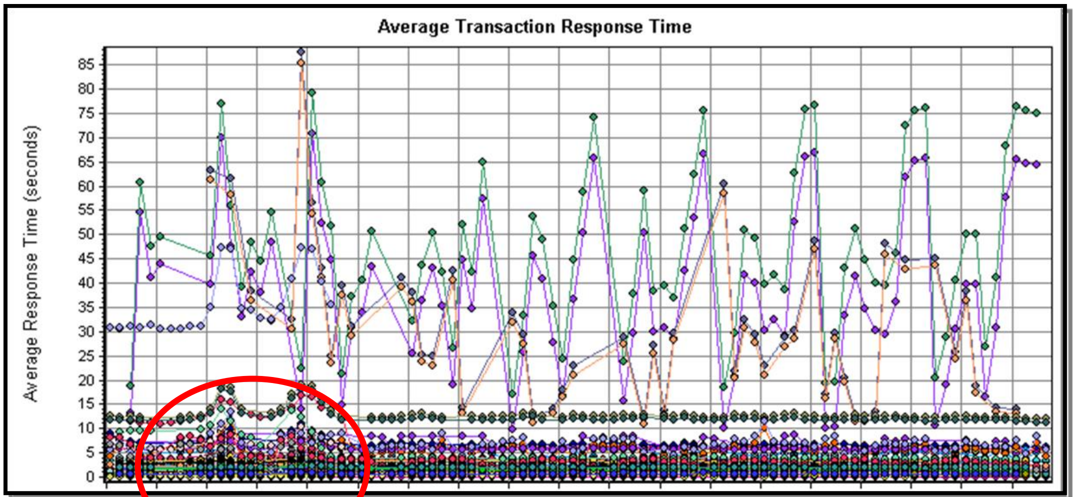


Figure 4-17, Server Response Times Affected

Pushing the system even harder with still higher login rate, CPU saturates at 100% at multiple time for extended duration (Figure 4-18).

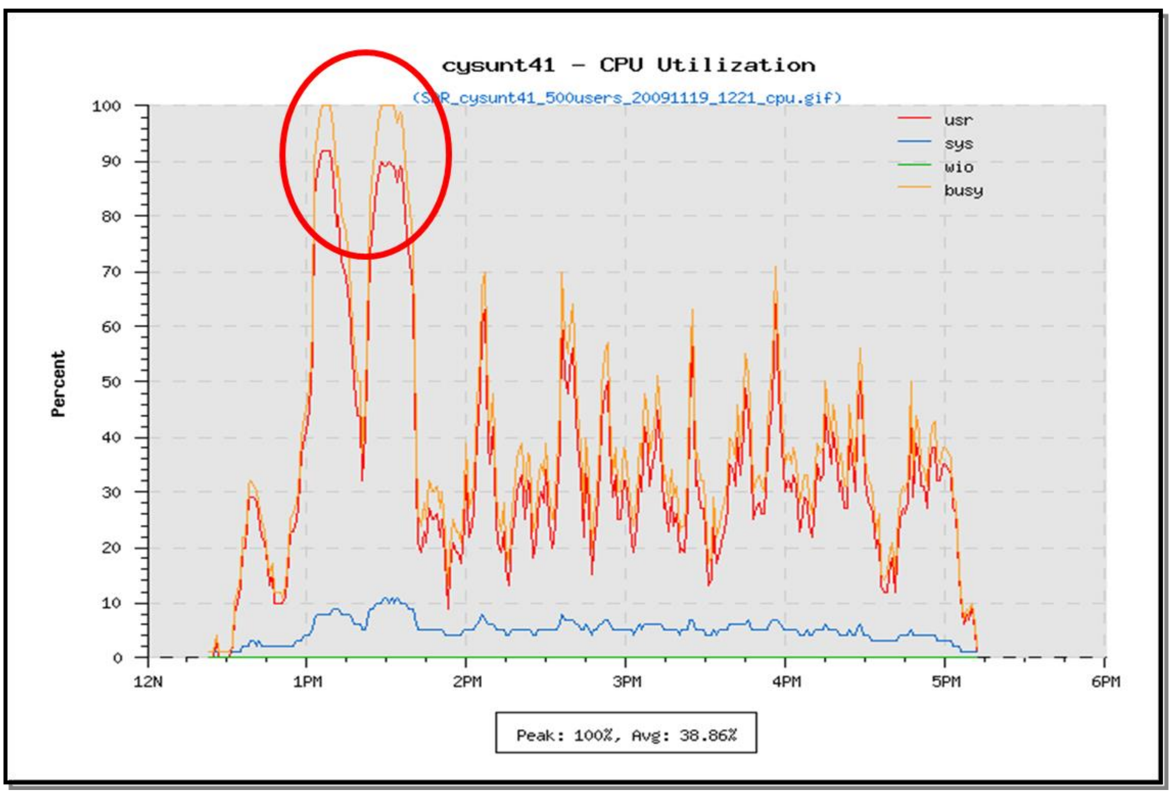


Figure 4-18, Sustained CPU Saturation

Although the runq-sz is still below the recommended maximum value of 1.52 times the number of CPUs, the *duration* of high runq-sz is much longer (Figure 4-19).

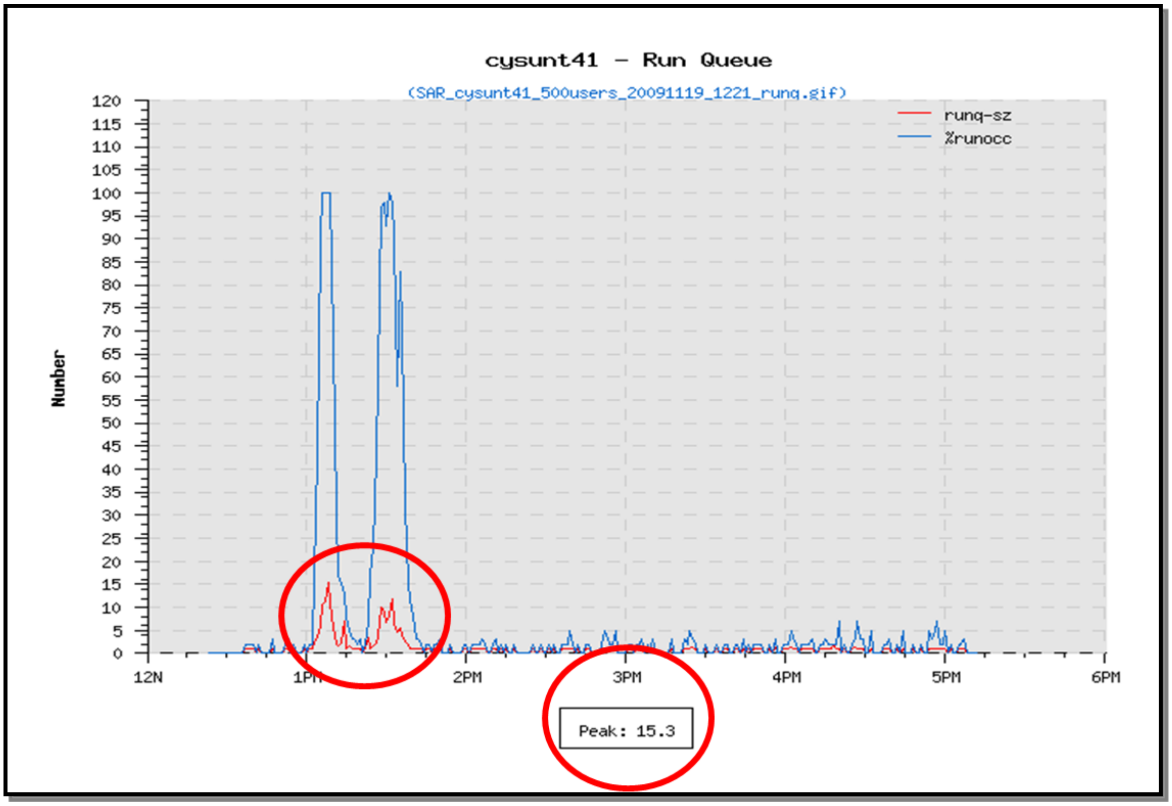


Figure 4-19, Sustained CPU Run Queue Size (runq-sz)

This can result in catastrophic failure; with response times regressing from seconds to minutes (Figure 4-20).

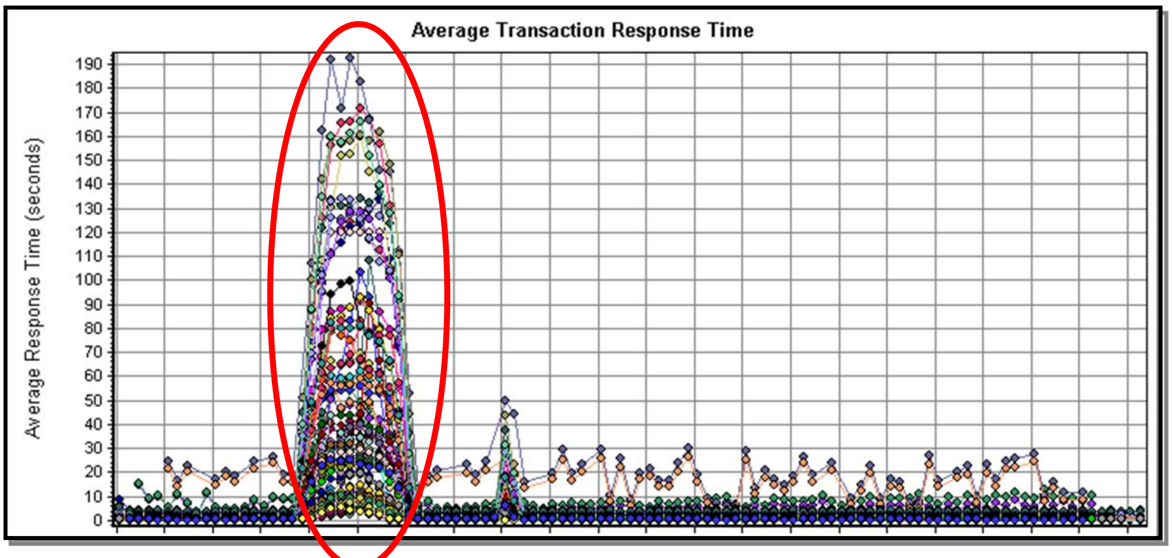


Figure 4-20, Catastrophic Server Response Time Impact

Key platform considerations to ensure adequate resources are:

- Keep CPU below 80- 95% and run queue size less than 1.5-2 times the number of CPUs (cores).
- Ensure free memory is at least 1 GB and page scans are zero, or very near zero.

Keep in mind that server resources are not the only factors that affect server response times. Similar effects can be caused by other conditions including misconfiguration of the server pool, web application server, JVM settings, and others.

4.5.4.1 Reducing Server Memory and/or Swap Consumption

The preceding charts were gathered with configuration settings that always provided the same tcserver process for the client sessions throughout the benchmark, thereby optimizing performance with maximum memory tcserver footprint and minimum CPU consumption. In those benchmarks, there was one assigned server process for each and every established user session in the web tier.

Memory demand can be reduced by configuring the local PROCESS_TARGET and/or global SOFT_TIMEOUT_STATELESS Server Manager parameter(s) such that tcservers will timeout after a certain idle time. It is reasonable to run at a 25% ratio of assigned tcservers to established web client sessions. This would significantly reduce memory demand as well as database login connections (by more than half), at the expense of slower response times and increased CPU demand to start replacement tcservers more often.

By default this timeout value is 20 minutes, and is expected to be a reasonable value for most production environments. However if you have a pool server that is short on memory but has plenty of CPU left, the number of simultaneous tcservers can be reduced by modifying the configuration as follows:

- First ensure that the Server Manager local PROCESS_TARGET profile maximum value is set to the desired maximum for the machine. If it is not, do so and monitor the machine to see if the number of assigned tcservers stays near that profile. If so, you need do nothing else.
- If not, you may also have to reduce the global SOFT_TIMEOUT_STATELESS timeout value below the default 20 minutes. Since this is a global parameter, it is configured on the web-tier.

Since this technique will increase the number of new tcserver process startups and assignments for web clients, you may also have to increase the Server Manager local PROCESS_WARM value to handle the higher bursts of initial assignments. There would be a higher number of such startups and assignments per user session. See section 4.3 above for additional information.

Note that this tuning will have no effect on the timeout of user sessions in the web-tier, and thus will not adversely affect the need for users to re-login to establish those sessions.

4.5.5 *Monitor System Usage*

Monitor system utilization regularly once the system is in pilot (and production) use. Usage profiles change as users find new methods to accomplish old tasks. Monitor systems regularly to flag future performance problems before they become critical. To ensure that memory has not become a bottleneck, check the total memory utilization (percentage) and virtual memory (VM) page-in and page-out rates regularly. For UNIX servers, use SAR and vmstat to monitor memory utilization and paging activity.

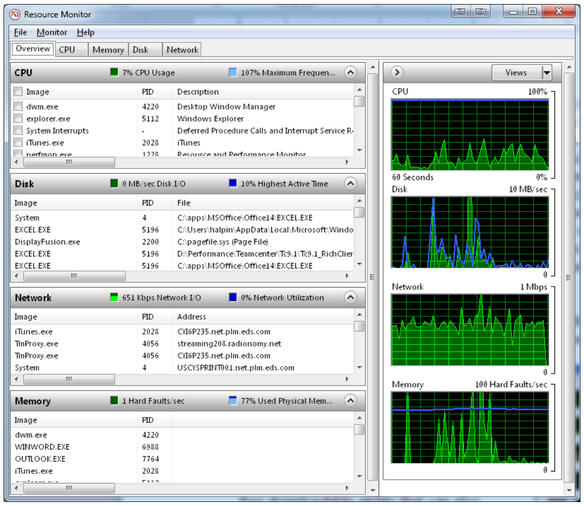


Figure 4-21, Resource Monitor

The Windows Task Manager and Performance Monitor tools provide rudimentary information on process execution, memory usage, handles, threads, and a number of other metrics. The Resource Monitor (Figure 4-21), introduced with Windows Server 2008 and Windows 7, is also good as it can provide rather detailed information for the overall system, memory utilization, disk and network I/O, as well as specific resources used for all instances of processes, and numerous other statistics.

Process Explorer (Figure 4-22), now available from Microsoft *Sysinternals*, is a free downloadable utility that can also provide summary performance information as well as process specific details.

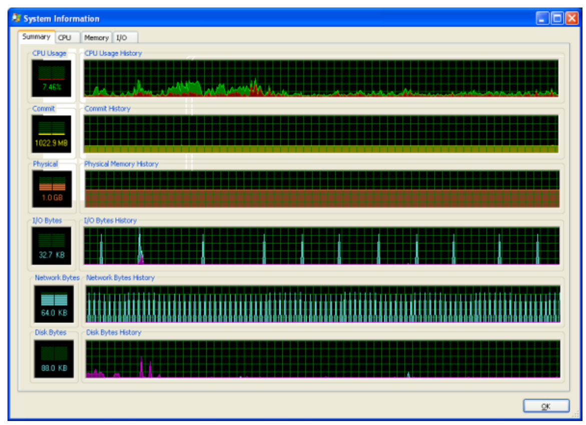


Figure 4-22, Process Explorer

Tools like HP OpenView and Sun Management Center have the added advantage of being able to trigger email or pager notification if system resources reach critical levels. HP OpenView can also be directly integrated with ARM compliant performance instrumentation in Teamcenter.

Use these tools frequently to monitor the health of system resources. As critical resources approach limits that may affect user response times, supplement the resources as required. This could include:

- Adding or upgrading CPUs or memory on server systems
- Adding additional servers are the Web Application and/or Enterprise tiers
- Adding disk space or upgrading to faster disks
- Adjusting operating system kernel or configuration parameters

- Moving the location of heavy users to be closer on the network to servers
- Rerouting network connections along different paths or adding network interface cards (NICs) to server machines.

Taking these precautionary measures may help eliminate excessive delays that cause user productivity to suffer.

4.6 Other Considerations

A number of operating system kernel parameters may need to be adjusted for large numbers of users. UNIX systems typically allocate tens or perhaps hundreds of handles per process, so the parameters limiting these may need to be increased. An experienced systems architect working closely with the Teamcenter system administrator should evaluate these recommendations in the context of the specific systems environment and usage profiles.

4.6.1 *Network Considerations*

Measurements with previous versions of Teamcenter have shown that load profiles of up to 1,000 logged-in users similar to those in appendix A, *Sample User Scenarios*, and appendix B, *Sample Usage Profiles* resulted in an average network utilization of less than 3 percent of 1 Gbit isolated LAN segment. See also section 10.2.2.4, *Network Resources*, for further information about network performance.

An excellent document with guidelines for Network Performance Tuning is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

4.6.2 *Operating System Considerations*

Enterprise servers spawn hundreds to thousands of tcserver processes, one for each logged in user plus each warm tcserver. On UNIX be certain the OS is configured to support the number of processes expected, or set to the maximum value allowed (65535 or ‘unlimited’), and the number of ephemeral ports are sufficient. The kernel parameter name(s) for the number of processes vary for each OS, e.g. for Solaris it is ‘ulimit’ (i.e. ‘ulimit –Hn’ for hard limit, ‘ulimit –n’ for soft limit). Similarly the parameter name(s) for the number of ports also vary for each OS, e.g. ‘tcpEndPort’.

4.6.2.1 BIOS settings

Processor Power States

Some processors support multiple power states intended to conserve energy when that is important. If performance is critical to your deployment, consider disabling C-States (processor states) and / or P-States (performance states) in the BIOS settings. Not all processors support these options, and not all BIOS implementations support configuring them. Refer to the hardware user's guide for your particular servers.

4.6.2.2 Windows Operating Systems

If you are upgrading from a 32 bit version of Teamcenter note there is no 32 bit version available for TC 11.2.1. Keep in mind for the 64 bit TC 11.2.1 there are a number of Windows and/or third party software parameters that may need to be adjusted for deployment or performance reasons.

For example, the 64 bit server pool manager may need additional Java memory, which might impact performance or cause Teamcenter to fail. You may need to increase the maximum Java heap size (-Xmx) for the JVM in which the pool manager runs.

For Microsoft Windows, consult your *Windows Server Reference Manual* or *System Administration Guide* for information about adjusting operating system parameters, or contact your platform provider for technical support.

For Windows systems, be certain the following **boot.ini** system settings are configured properly:

- **/MAXMEM** is not set to a low number causing Windows to page excessively.
- **/BURNMEMORY** is not set.
- **/ONECPU** and **/NUMPROC** are not set limiting the number of CPUs used.
- **/NODEBUG** is set.

The registry settings for **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters** shown in Table 4-7 are known to improve the performance of data transfers. (AFD is the driver that handles Winsock).

Table 4-7, Local Machine Parameters

Key	DWORDS (Values in Decimal)
DefaultReceiveWindow	16385
DefaultSendWindow	16384
InitialLargeBufferCount	22
InitialMediumBufferCount	30
LargeBufferSize	8192
MediumBufferSize	4096

The sizing parameter values are only guidelines for Enterprise servers. An experienced systems architect working closely with the system administrator should evaluate these recommendations in the context of your specific systems environment and usage profiles.

A number of network parameters shown in Table 4-8 were found to improve server response times and/or ability to scale to very large number of users with Windows 2008 R2. In general, these should be set on all server tier machines (i.e., Web, Enterprise, Resources) if several thousand concurrent users are planned. Again, an experienced systems architect working closely with the system administrator should evaluate these recommendations in the context of your specific systems environment and usage profiles.

Table 4-8, Windows Server 2008 R2 Network Parameters

Parameter	Value (Values in Decimal)	Location / Command / Purpose
TcpTimedWaitDelay	0x1E (30s)	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\(REG_DWORD)
maxuserport	0x0000ffff (65535)	Default value is 0xF0, which sets the wait time to 240 seconds (4 minutes), effectively holding the connection hostage for 4 minutes, which increasing the number of connections in use. 30s is a more reasonable value.
Receive-Side Scaling State	disabled	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\(REG_DWORD)
Chimney Offload State	disabled	This is especially important on the web tier as each user requires a separate ephemeral IP port.
NetDMA State	disabled	netsh int tcp set global rss=disabled
Direct Cache Access (DCA)	disabled	netsh int tcp set global chimney=disabled
Receive Window Auto-Tuning Level	normal	netsh int tcp set global netdma=disabled
Add-On Congestion Control Provider	ctpc	netsh int tcp set global dca=disabled
ECN Capability	disabled	netsh int tcp set global autotuninglevel=normal
RFC 1323 Timestamps	disabled	netsh int tcp set global congestionprovider=ctcp

4.6.2.1 Trend OfficeScan Limitations

If Trend OfficeScan is running on the server pool machine, the OfficeScan tmtdi.sys driver may present a bottleneck under high load that will cause random transaction failures. Trend has been notified of this issue. Until Trend corrects the issue or replaces the tmtdi.sys driver with a different mechanism, the number of concurrent tcserver processes (users) per server pool machine may need to be limited. Configure multiple machines as required to support the total expected concurrent user count.

How many concurrent users can be supported per machine depends on the production usage profile; some experimentation may be required to determine the maximum number that can be supported. In Siemens PLM benchmarks with the APA standard usage profile, about 750 could be supported.

Chapter

5 *Managing Database Server Deployments*

Go to main Table of Contents

Chapter

5 *Managing Database Server Deployments*

This chapter focuses on configuration, sizing, and maintenance of Oracle and Microsoft SQL Server Databases.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- Server installation for: *Windows*, *UNIX/Linux*
- *Maintaining Database Servers*
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

This chapter is organized as follows:

Section 5.1	*General Teamcenter Database Performance Guidelines*
Section 5.2	*Managing Oracle Database Servers*
Section 5.2.1	*Oracle Performance Tuning Quick Reference*
Section 5.2.2	An in-depth review of *Oracle Server Performance* issues and diagnosis, and configuration and tuning guidelines. The Oracle configuration accounts for a large portion of customer performance inquiries and tuning an Oracle instance is a fairly complex subject.
Section 5.3	*Managing Microsoft SQL Server Database Servers*
Section 5.3.7.2	*SQL Server High Availability*

5.1 General Teamcenter Database Performance Guidelines

Achieving optimal performance from a database server is an iterative process requiring careful record keeping and patience that involves measuring, making configuration changes, and measuring again. However, there are several options and general practices that can have a significant impact on overall Teamcenter performance. Database performance issues most likely to be encountered fall into the following categories:

- Properly sizing the server
- Database configuration and maintenance
- Index maintenance
- Tablespace/data file organization

Properly sizing the server, configuration and maintenance, and proper index maintenance tend to have the greatest impact on database performance for Teamcenter. A missing or out-of-date index can cause an operation to take tens of minutes rather than seconds. A few key parameter settings and maintenance scripts can have dramatic impact on query performance, especially for Oracle. Growth should be planned for from the beginning and monitored frequently on any production environment with regards to processor, memory, and file I/O utilization. All of these factors (especially file I/O) as well as disk recoverability can be affected by the careful placement of data and log files used by the Teamcenter database as well as database system files.

5.1.1 *Properly Sizing the Server*

Configuring a database server for performance is much like tuning any other server; CPU, memory, disk, and network resources must be adequately sized and configured.

NOTE

Because Teamcenter can consume significant database resource **Siemens PLM Software recommends that no other applications are served by the database server**. If other database applications ***must*** be served from the same machine as Teamcenter, there should be a dedicated database instance for Teamcenter. This allows the designated instance to be tuned specifically for Teamcenter.

5.1.1.1 CPU Resources

Like all other Teamcenter components, database CPU utilization should be kept below 80–90 percent. Above that level, response times degrade more rapidly. However, an improperly configured or tuned database can consume much more CPU than would otherwise be required. If CPU utilization is above 80 percent, and the CPU process queue[44] exceeds 2x the number of CPUs, you may have a CPU bottleneck. However, it is just as likely that tuning the database instance will reduce CPU consumption.

On UNIX database servers Oracle employs numerous processes for its operation. These processes are fairly autonomous and thereby actually compete with each other for CPU resource. Although a properly sized single processor UNIX system may perform acceptably with 100 or fewer Teamcenter users, Oracle tends to perform better on multiprocessor UNIX systems.

SQL Server on the other hand, as well as Oracle on a Windows platform, takes better advantage of Windows' threading model, so fewer, faster processors are generally adequate.

5.1.1.2 Memory Resources

Like other applications, a database manager performs better if its programs and data are always in physical memory. Oracle, SQL Server, and DB2 all try to keep as much of the database as possible in an in-memory data buffer and move data to/from disk according to the criteria you specify in various initialization parameters. If insufficient memory is allocated to the data buffer, the database accesses disk more frequently and performance suffers. If you allocate more memory than is available, the operating system begins to page, and again performance suffers.

To ensure that memory has not become a bottleneck, check the total memory utilization (percentage) and virtual memory (VM) page-in and page-out rates regularly. For Oracle or DB2 on UNIX servers, use SAR and vmstat to monitor memory utilization and paging activity, or Perfmon on Windows. Microsoft SQL Server memory can be monitored using PerfMon memory counters for the SQL Server process (sqlservr) and the SQL Server: Memory Manager (Target Server Memory and Total Server Memory counters); see Figure 5-1.

[44] CPU Process Queue is the number of processes that are waiting for the CPU to resume execution. This does not include processes that are executing, waiting (memory, I/O, streams…), or sleeping.

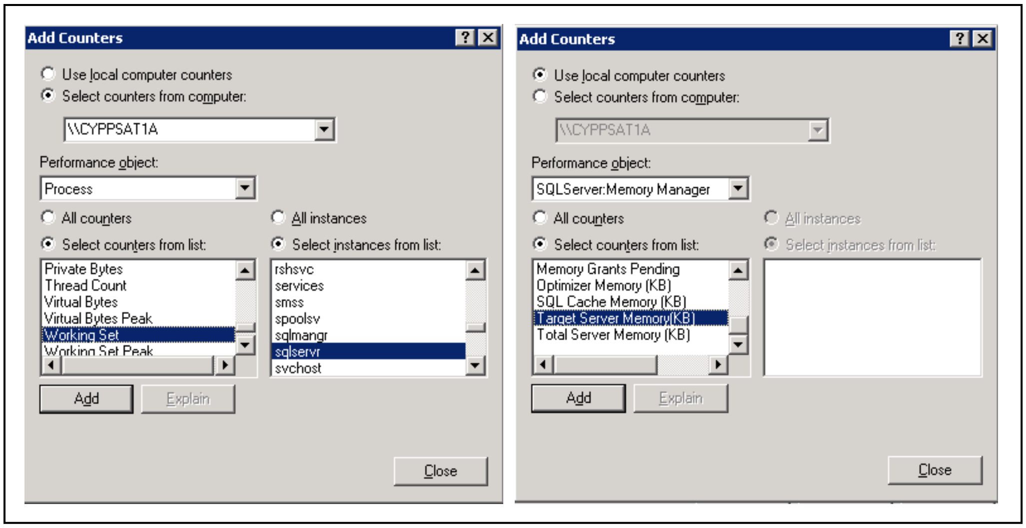


Figure 5-1, PerfMon Memory Counters for SQL Server

Memory sizing is something of a balancing act. Properly balanced, physical memory should be nearly 100 percent utilized, with little or no disk paging (as opposed to normal disk I/O). This may require adding physical memory to your database server.

Unlike other types of UNIX servers, database servers require little Operating System Buffer Cache[45]. Set OS buffer cache on UNIX systems to the minimum recommended by the OS supplier, usually 5–10 percent of total memory.

5.1.1.3 Disk Resources

Even with properly balanced database memory, disk I/O on a database server can be significant. Accessing data from disk is very time consuming and every effort should be made to reduce disk access times. Disk read / write latencies should be 5ms or less and remain relatively consistent throughout the day. There are a number of measures you can take to optimize disk subsystem performance.

- For large deployments, high throughput low latency Fiber Channel based SAN file servers are highly recommended. Benchmarks have shown that for high load usage profiles, internal disk arrays may not be able to keep up, even with properly configured RAID arrays.
- Consider multiple smaller, fast disk drives, rather than a few large drives. Data can then be spread across multiple drives. Spreading data across drives allows the OS to perform multiple drive operations at the same time, improving throughput. How many drives are appropriate is dependent on the type of drive, interface, and controller; consult your hardware provider for guidance.

[45] OS buffer cache optimizes disk I/O by keeping recently used data blocks in memory so that subsequent accesses to the same data are satisfied faster. Because Oracle manages its own buffer cache (the SGA), the OS buffer cache is redundant, and can actually impair performance by moving data twice, once from Oracle SGA to OS cache, then to disk (or vice versa).

- For internal drives consider multiple disk controllers with several drives on each. This allows data to be spread across controllers as well, allowing data to be read and written in parallel. As with drives, consult your hardware provider for guidance about how many controllers are recommended.
- For internal drives consider RAID configurations that improve (or at least do not degrade) throughput. For additional information on RAID, see the next section on *RAID Configurations*.
- Most platform suppliers now offer some form of Virtual File System (VFS) capabilities. These usually include the ability to stripe multiple disks together to increase throughput, and to mirror data on multiple disks to provide redundancy. Consult your platform system documentation to determine the capabilities available.
- For internal drives consider putting temporary data file(s)[46] on a separate controller from the Teamcenter database files. Temporary data files are the database's work area used for sorting, sub-queries, aggregates, etc., and although small should be sized to allow rapid growth. The data within temporary data files is mostly transient and would not benefit from disk redundancy.
- Oracle, SQL Server, and DB2 store essential database definition information regarding logins, devices, and other information in special tables/files. This is critical to the system and is required for recovery. For example, in Oracle this is the 'system' tablespace; in SQL Server, the 'master' database. In addition to regular backups, these data files should be placed on a fault tolerant file system for recoverability.
- For very large deployments consider high capacity / throughput optimized file systems such as the FAS series from NetApp with optional Flash Cache performance accelerator. These servers have been shown to support up to 10,000 concurrent Teamcenter users, while accelerators like the Flash Cache provide FC drive level performance with less expensive SATA drives.

5.1.1.4 RAID Configurations

As the cost of disk RAID configurations continues to become more affordable, customers are implementing them for Oracle systems. Which RAID option to use is a balance of the 1) cost of the disks vs. 2) the cost of the system being down vs. 3) performance. There are a number of documents in the public domain describing and comparing RAID configurations which are available at the following URLs:

- *http://en.wikipedia.org/wiki/RAID*
- *http://www.systemlogic.net/articles/01/1/raid*

[46] TEMP tablespace for Oracle, tempdb data file(s) for SQL Server

NOTE

Although most high capacity optimized file systems such as the NetApp FAS series implement some form of redundant RAID, they perform so well that the following considerations can generally be disregarded.

In general, *RAID 0 (striping)* gives the best performance, but no fault tolerance. In this configuration, a failure of a single drive renders the data on all drives in the stripe set inaccessible. Institute a strict backup procedure for RAID 0 volumes.

RAID 1 (mirroring) provides fault tolerance, but may show slower performance on write operations. With RAID 1, intelligent controllers or controller drivers can access less busy drives for reading data, but writing data may be delayed by writing to two disks depending on the hardware implementation.

RAID 3 (bit interleaved parity) provides some fault tolerance, but is not in general use.

RAID 5 (block interleaved, distributed parity) is highly fault tolerant. RAID 5 exhibits RAID 0 performance for reads, but can show poor performance for writes depending on the hardware implementation. Although newer RAID controllers have improved write performance, for many controllers writes to RAID 5 drive sets require 4 sequential I/O operations (2 reads, 2 writes) instead of 1.

RAID 0+1 (striped then mirrored) is a combination of RAID 0 and RAID 1. *RAID 1+0* is a variation where the array is mirrored then striped. RAID 0+1 (or 1+0) provides excellent read and write performance, but are the most expensive configuration. For additional information on this configuration, see also the techweb site above.

In general, RAID usually impacts write operations more so than read operations. This is especially true where parity need to be calculated (for example, RAID 3 or RAID 5). **You should not use RAID 5 for Oracle redo log files or SQL Server transaction log files** unless the hardware provider guarantees write performance**.** Online or archived redo log files can be put on RAID 0 devices, or if fault tolerance is important, on RAID 1 devices. *TEMP* tablespace / tempdb data files should also go on RAID 0 or 1 rather than RAID 5. The reason for this is that streamed write performance of distributed parity (RAID 5) is not as effective as that of simple mirroring (RAID 1) in most hardware implementations.

Operating system swap space can be placed on RAID devices without affecting Oracle.

Table 5-5 provides relative rankings for RAID configurations for specific Oracle file types. The rankings range from 1 (best) to 5 (worst)[47].

[47] Adapted from the Oracle Enterprise 8i Performance Tuning Workshop

Some database suppliers have specific recommendations for various types of RAID. Refer to **RAID Configurations** in the database specific section later in this chapter.

5.1.1.5 Network Resources

The network factor that most affects database performance for Teamcenter is *latency*. For many operations, Teamcenter makes multiple queries to the database server to make the user experience more interactive with more real-time feedback. Each query and result must traverse the distance between client and server. If 100 queries are made across a 200 ms WAN network link, the overall operation requires 20 seconds to complete. This may be unacceptable to the end user. That same operation on a 1 ms LAN requires only 1/10 of a second, which is imperceptible to the user.

Performance is noticeably slower if the network latency exceeds 5–10 ms between tcserver process and database server.

Remember that latency is determined by distance and the number of hops between the client and server; there is little that can be done to reduce latency. Teamcenter performance is also impacted by the network bandwidth of the database connection or high network utilization, but latency is the largest overall network factor for database servers.

An excellent document with guidelines for Network Performance Tuning is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

See also section 10.2.2.4, *Network Resources*, for further information about improving network performance.

For optimal performance, Siemens PLM Software recommends that Teamcenter application servers[48] reside on the same Gbit LAN as the database server.

5.1.2 *Database Configuration and Maintenance*

Configuration and maintenance are generally peculiar to a type of database management system. Refer to the sections later in this chapter for specific Oracle, SQL Server, and DB2 guidelines.

5.1.3 *Index Maintenance*

Indexes are used for searches to avoid searching through the entire tables (full table scans) and during disk sorts. Properly created and maintained database indexes

[48] Web, Enterprise, FMS

undoubtedly have the largest single impact to performance. Missing, inefficient, out-of-date, or corrupt indexes can cause operations to take minutes, even hours, when they should complete in seconds. Teamcenter creates the required indexes at installation for the majority of customers. Still, if the database is not using them or they are out-of-date performance suffers. There are three steps to ensure that indexes are effective:

- The correct indexes are present.
- The database is using the indexes.
- Indexes are properly maintained.

5.1.3.1 Optional Indexes

Experience from the field has shown some deployments that use variant expressions on structures can be slow depending on the data content and expressions used. If expanding structures with variant expressions in Structure Manager is slow in your environment, consider adding the following index, and then measure the impact to response times.

```
create unique index PIPVARIANTEXPRE_1 on
PVARIANTEXPRESSION(PFORMULA_STATE,POPERATOR,PUID);
```

If performance is negatively impacted by the addition of this index, or has no impact, then remove it, as the overhead of maintaining it would not be warranted:

```
drop index PIPVARIANTEXPRE_1;
```

5.1.3.2 Verifying Teamcenter Indexes

To determine whether all the default Teamcenter indexes exist in the database, use the **index_verifier** utility. The verifier ensures all indexes created at installation/upgrade time are in place, including any indexes you created with the install utility. It does this by consulting the internal Teamcenter data dictionary (**pom_indices**), which defines all indexes created by the install utility. For detailed instructions for using the index verifier, see the *Teamcenter Online Help Collection*.

When executed, the verifier examines each Teamcenter object class and checks for missing indexes. If it finds missing indexes, it reports them along with an SQL statement required to create the index. Without errors, the output from **index_verifier** is a series of lines of the form:

```
CHECKING CLASS [<class name>] FOR MISSING INDEXES
```

Run the index_verifier utility with the -o=DRYRUN argument to output the missing indexes to a file, for example:

```
index_verifier -u=username -p=password -g=group
-o=DRYRUN > index_verifier_dryrun.txt
```

If missing indexes are found, there will be output similar to:

```
REM Found missing functional index on table
<PMEAPPEARANCEPATHNODE> columns <rparentu>
```

Creating Missing Indexes

If missing indexes are found from –o=DRYRUN iteration, run index_verifier utility with the -o=DO_IT argument to find the missing indexes and create the replacement indexes in one step, for example:

```
index_verifier -u=username -p=password -g=group -o=DO_IT
```

NOTE

Do not use the **install –add_index** function to add indexes reported as missing by the **index_verifier** utility; they are already in the Teamcenter POM data dictionary so the **index_verifier** utility already knows about them. Use the create index statements from the index_verifier instead.

The **index_verifier** utility also detects any missing custom indexes that were created using **install –add_index** (because the install utility adds them to the Teamcenter data dictionary). The **index_verifier** utility cannot check the existence of indexes you create with SQL; those are not added to the data dictionary. For the **index_verifier** utility to find all missing indexes, always create indexes (except those reported by the **index_verifier** utility) using the installation utility as follows:

```
install -add_index infodba <password> dba index_name
unique_option class attr1 attr2 ..."
```

For detailed installation utility instructions, see Teamcenter *Help Collection – Administering Teamcenter – Utilities Reference – Maintenance utilities – Installation*.

NOTE

Always rerun the **index_verifier** utility after a Teamcenter upgrade.

5.1.3.3 Index Creation

Generally indexes do not need to be manually created unless the Teamcenter data schema has been extended with additional classes/properties and users search for objects using the added properties. Also, custom queries against properties not already indexed during install/upgrade may require new indexes.

Additional Indexes

At installation/upgrade time, Teamcenter creates indexes on the properties that Teamcenter typically queries. For example every table in Teamcenter has at least an index on the primary key (PUID).

Extended Data Model Indexes

If the Teamcenter data schema is extended by adding classes or objects, or by adding properties to them, indexes on these properties may need to be created. When you add an attribute to an existing class via **install –add_attr** or any other Teamcenter/POM tool, indexes are not added for this property because there is no need to have indexes on properties unless a custom (saved) query is created against that property. Exercise restraint when creating indexes to avoid creating indexes on attributes unnecessarily.

Saved Queries

Similarly, if custom (saved) queries or user-exit queries are created that query one or more properties (custom or not) that are not indexed, then indexes need to be created on these properties. Not defining the necessary indexes for your new saved query could impact performance significantly. However, although indexes help improve query performance, too many of them may have a negative effect on save/edit performance. This is true when the index is defined on an attribute that gets updated frequently. Take care not to add indexes unnecessarily. Unnecessary indexes impair the performance of database changes (creates, updates, deletes).

Determining Additional Needed Indexes

So the question is how to know what indexes to add? This is not easy answer, because not all attributes involved in the query are known to the user who defines the query. In general, additional indexes may be needed if specific SQL queries, or newly created customer queries execute slowly.

Slow SQL Reporting

In order to help identify slow SQLs (either requiring optimization or due to missing indexes) Teamcenter can report SQL statements that take longer than a time you specify with an environment variable (**TC_SLOW_SQL**). With **TC_SLOW_SQL** set with an Oracle or SQL Server implementation, Teamcenter not only reports slow queries, it also includes the query plan (which explains the access path for obtaining the data) for the first 10 queries reported (and, on Oracle, even installing the plan table if it is missing).

NOTE

In large deployments where the database server is heavily loaded, the **TC_SLOW_SQL** reporting can place additional load on the system such that the reporting itself will begin to impact performance. This can result in even more **TC_SLOW_SQL** reporting that causes the database load to escalate indefinitely. Only use **TC_SLOW_SQL** as a diagnostic tool; always disable **TC_SLOW_SQL** in a production system by setting **TC_SLOW_SQL=-1**.

To enable slow SQL reporting set the value of **TC_SLOW_SQL** to a floating point number greater than zero (e.g., **TC_SLOW_SQL=7.5**). For example if you set this to **1** or **1.0**, all

SQLs that take more than 1.0 second are reported in the syslog along with the explain plan.

For any SQL that takes more than the time specified, the full SQL text is reported in the syslog, the execution plan is then reported, and information about tables and indexes used by the query is listed, including statistics. With the explain plan, you can see exactly how the database went about accessing tables and indexes.

The *explain plan* reported in the syslog helps you:

- Determine if any *full table access* operations were executed (full table scans)
- Determine whether indexes are missing
- Analyze the table/index statistics

NOTE

The actual method for performing this analysis varies between Oracle and SQL Server. Please follow the appropriate section for your database below. In SQL Server, the execution plan within *SQL Query SQL Analyzer* provides a visual representation of the optimizer plan.

Slow SQL Analysis: SQL Server Example

In an SQL Server implementation, text similar to the following would be found in the syslog (with TC_SLOW_SQL set to a low value, such as 0.2):

SQL Statement

```
SELECT t_01.puid, t_01.pproperty_name, t_02.ptype_name,
t_01.VLA_277_3 AS VLALENGTH FROM PPROPERTYINFO t_01 , PIMANTYPE
t_02 WHERE ( t_01.rbusiness_objectu IN  ('gOLAAAAkYIqm6C',
'xCBAAAAkYIqm6C', 'RyDAAAAkYIqm6C', 'RyHAAAAkYIqm6C',
'xCFAAAAkYIqm6C', 'xNFAAAAkYIqm6C', 'xNJAAAAkYIqm6C' ) AND (
t_01.rbusiness_objectu = t_02.puid ) );
```

===>Took .219 seconds to execute that SQL

Explain Plan

The execution plan follows the query (given above) in the syslog file. The plan has been reformatted here for readability (with extra digits trimmed), but the actual syslog output is very similar.

Subtree Cost	Est. IO	Est. Cpu	Est. Rows	Statement Text
0.1492	0	0	34	`SELECT t_01.puid, t_01.pproperty_name, t_02.ptype_name, t_01.VLA_277_3 AS VLALENGTH FROM PPROPERTYINFO t_01 , PIMANTYPE t_02 WHERE ( t_01.rbusiness_objectu IN ('gOLAAAAkYIqm6C', 'xCBAAAAkYIqm6C', 'RyDAAAAkYIqm6C', 'RyHAAAAkYIqm6C', 'xCFAAAAkYIqm6C', 'xNFAAAAkYIqm6C', 'xNJAAAAkYIqm6C' ) AND ( t_01`
0.1492	0	0.0001	34	`Nested Loops (Inner Join, OUTER REFERENCES:([t_01].[rbusiness_objectu], [Expr1004]) WITH UNORDERED PREFETCH)`
0.1028	0.0935	0.0093	35	`Clustered Index Scan (OBJECT:([tc].[dbo].[PPROPERTYINFO].[PIPPROPERTYINFO] AS [t_01]), WHERE:([tc].[dbo].[PPROPERTYINFO].[rbusiness_objectu] as [t_01].[rbusiness_objectu]='RyDAAAAkYIqm6C' OR [tc].[dbo].[PPROPERTYINFO].[rbusiness_objectu] as [t_01].[rbusiness_objectu]='RyHAAAAkYIqm6C' OR [`
0.026	0.0031	0.0002	1	`Clustered Index Seek (OBJECT:([tc].[dbo].[PIMANTYPE].[PIPIMANTYPE] AS [t_02]), SEEK:([t_02].[puid]=[tc].[dbo].[PPROPERTYINFO].[rbusiness_objectu] as [t_01].[rbusiness_objectu]), WHERE:([tc].[dbo].[PIMANTYPE].[puid] as [t_02].[puid]>='RyDAAAAkYIqm6C' AND [tc].[dbo].[PIMANTYPE].[puid] as [`

Analyzing the Explain Plan: SQL Server

The plan is an ordered list of the operations performed to execute the SQL statement. These are usually shown in a nested form to convey the order of operations that provide the input for the next steps, with the most deeply nested operations performed first (with each controlled by the operation just above it). In the example plan given above, the following operations are performed against the indicated tables:

Operation	Tables	Notes
`Nested Loops`	`PPROPERTYINFO (t_01)`	*Loop over the* `rbusiness_objectu` *values in the list, performing each of the two inner steps*

Operation	Tables	Notes
Clustered Index Scan	PPROPERTYINFO (t_01)	*Find rows in* PPROPERTYINFO *with* rbusiness_objectu *that matches one of the values in the list*
Clustered index Seek	PPROPERTYINFO (t_01), PIMANTYPE (t_02)	*Find the rows in* PIMANTYPE *with* puid *equal to the* rbusiness_objectu *value*

In this case, SQL Server works its way through the values in the IN list, finding the matching rows in PPROPERTYINFO then in PIMANTYPE.

SQL Server makes use of many different plan strategies, most of which are beyond the scope of this document. (Refer to the SQL Server documentation or related books for more information.) However, the methods in this particular plan merit explanation.

The "Clustered Index Scan" is what other databases consider a table scan. The PPROPERTYINFO table is declared to be a *clustered* table when originally installed, which means data is organized physically based on the clustering key (puid in this case). This arrangement is very similar to an index, which explains the name of the method. In this particular case, SQL Server is examining all the rows in PPROPERTYINFO in order to find those with rbusiness_objectu matches an individual value. As can be seen in the query plan, this is the single most expensive part of this query.

The "Clustered Index Seek" is the straightforward use of an index. Having located all the PPROPERTYINFO rows with a given rbusiness_objectu value, SQL Server then used the index PIPIMANTYPE on the PIMANTYPE table to find the single row (since PIPIMANTYPE is a unique index) with puid matching the input value from the nested loops. From the plan, this is a quick operation that is expected to return a single (1) row each time.

The "Nested Loops" step simply invokes the preceding two operations repeatedly for each value in the list.

Determining the Missing Index: SQL Server

In looking at this particular query, the slowest part is, again, the clustered index scan, which is effectively a full table scan for each input value. Adding an index on the rbusiness_objectu column (which corresponds to the business_object property of the PROPERTYINFO class) will likely speed up this query.

There are two ways to add this index. The first (and recommended) method is to use the `install` utility with the `-add_index` option as described earlier in this document. Using this utility will ensure that Teamcenter registers the index and that the index verifier can check the index when requested.

```
install -add_index infodba infodba dba MYCORP_PIPROP_RBUSU 0 \
  PROPERTYINFO business_object
```

Parameter	Significance
`-add_index`	Indicates an index is to be added to the model
`infodba`	Username of the Teamcenter administrator
`infodba`	Password of the Teamcenter administrator
`dba`	Group associated with the Teamcenter user (usually dba)
`MYCORP_PIPROP_RBUSU`	The name of the index within the POM data model. (Not used at the database level.) A prefix to identify local additions versus default indexes is recommended.
`0`	0 = non-unique index; 1 = unique index
`PROPERTYINFO`	The POM class
`business_object`	The POM class attribute to be indexed (attribute name, not database column name)

The second method simply creates the index directly, using, for example, Microsoft SQL Server Management Studio. Using the studio, SQL similar to the following can be entered and executed to construct the index:

```
USE tc
go
CREATE INDEX [MYCORP_PIPPROPERTYINFO_0]
ON [dbo].[PPROPERTYINFO] ( rbusiness_objectu )
go
```

As in the preceding method, it is recommended that a site-specific prefix (MYCORP in this sample) be added to the index name to make it easy to spot local additions to the database. Also, unlike the install utility, a command to create the index directly references only database schema objects, not POM objects. By default, SQL Server indexes are non-clustered, and this is the recommended setting for site-specific indexes. If the indexed columns are unique, the `UNIQUE` keyword can be added to the `CREATE INDEX` command.

Once the new index is created, SQL Server will automatically gather statistics on it and evaluate it for use in queries. In the earlier example, adding the index on a sample system saw the index scan replaced with an index seek, with significantly lower execution times. The statement no longer appeared in the syslog as a slow SQL statement.

Slow SQL Analysis: Oracle Example

SQL Statement

```
SELECT DISTINCT t_02.PUID FROM PAPPEARANCEATTR t_01, PAPPEARANCE t_02
  WHERE ( ( t_02.rappearance_rootu = 'A180i4zyAAAMeC' )

AND ( ( t_01.pvalue LIKE '3%' AND ( t_01.rdefinitionu ='AN5w9yFrAAAMeC
' ) )
    AND ( t_02.PUID = t_01.rappearanceu ) ) )

INTERSECT SELECT t_04.PUID from PAPPEARANCEATTR t_03, PAPPEARANCE t_04
  WHERE ( ( t_03.pvalue LIKE '4%' and ( t_03.rdefinitionu='AN8w9yFrAAA
MeC' ) )
    AND ( t_04.PUID = t_03.rappearanceu ) );

===>Took 1.016 seconds to execute that SQL
```

Explain Plan

```
SELECT STATEMENT
INTERSECTION
 SORT   UNIQUE
   HASH JOIN
     TABLE ACCESS    BY INDEX ROWID PAPPEARANCE
       INDEXRANGE SCAN      PIPAPPEARANCE_1
     TABLE ACCESS    FULL   PAPPEARANCEATTR
 SORT   UNIQUE
  NESTED LOOPS
     TABLE ACCESS    FULL   PAPPEARANCEATTR
     INDEX UNIQUE SCAN     PIPAPPEARANCE
```

Tables & Indexes (Oracle only)

```
table PAPPEARANCE has 14493 rows, last analyzed 2002/07/10 with 7014
rows
index PIPAPPEARANCE on PAPPEARANCE (PUID) last analyzed never over 0
rows
index PIPAPPEARANCE_1 on PAPPEARANCE (RAPPEARANCE_ROOTU) last analyzed
never over 0 rows
index PIPAPPEARANCE_2 on PAPPEARANCE (RCOMPONENT_ITEMU) last analyzed
never over 0 rows
index PIPAPPEARANCE_3 on PAPPEARANCE (RCOMPONENT_BOM_VIEWU) last
analyzed never over 0 rows
index PIPAPPEARANCE_4 on PAPPEARANCE (ROCC_THREADU) last analyzed
never over 0 rows
index PIPAPPEARANCE_6 on PAPPEARANCE (RPARENTU) last analyzed never
over 0 rows
table PAPPEARANCEATTR has 81744 rows, last analyzed 2002/07/10 with
47514 rows
index PIPAPPEARANCEAT_2 on PAPPEARANCEATTR (PUID) last analyzed never
over 0 rows
index PIPAPPEARANCEAT_3 on PAPPEARANCEATTR (RAPPEARANCEU) last
analyzed never over 0 rows
index PIPAPPEARANCEAT_4 on PAPPEARANCEATTR (RAPPEARANCEU,
RDEFINITIONU) last analyzed never over 0 rows
table PAPPEARANCEATTR has 81744 rows, last analyzed 2002/07/10 with
47514 rows
index PIPAPPEARANCEAT_2 on PAPPEARANCEATTR (PUID) last analyzed never
over 0 rows
index PIPAPPEARANCEAT_3 on PAPPEARANCEATTR (RAPPEARANCEU) last
analyzed never over 0 rows
index PIPAPPEARANCEAT_4 on PAPPEARANCEATTR (RAPPEARANCEU,
RDEFINITIONU) last analyzed never over 0 rows
```

As you can see, there is a wealth of information produced for this sample query.

1. The SQL statement generated by Teamcenter.
2. An explain plan.
3. Information about tables and indexes involved in the query including:
    a. Number of current rows on each table
    b. Last time tables and indexes were analyzed and how many rows they had at the time

The following sections describe how to interpret this data.

Analyzing the Explain Plan: Oracle

The explain plan is an ordered, indented list of the operations performed to execute the SQL statement. It is formatted similar to the following:

SELECT STATEMENT

```
INTERSECTION
 SORT   UNIQUE
   HASH JOIN
     TABLE ACCESS    BY INDEX ROWID PAPPEARANCE
       INDEXRANGE SCAN      PIPAPPEARANCE_1
     TABLE ACCESS    FULL   PAPPEARANCEATTR
 SORT   UNIQUE
  NESTED LOOPS
     TABLE ACCESS    FULL   PAPPEARANCEATTR
     INDEX UNIQUE SCAN     PIPAPPEARANCE
```

In this sample explain plan, there are three table accesses:

```
    TABLE ACCESS    BY INDEX ROWID PAPPEARANCE
    TABLE ACCESS    FULL   PAPPEARANCEATTR
    TABLE ACCESS    FULL   PAPPEARANCEATTR
```

one *indexed* table access (**BY INDEX ROWID**), and two *full table scan* accesses (**TABLE ACCESS FULL**).

This means the database had to read (scan) the entire **PAPPEARANCEATTR** table twice. **Full table scans are very I/O intensive and cause the query to take longer than if an indexed access can be made.**

Determining the Missing Index: Oracle

Knowing the table is the first step, which was identified in the explain plan (**TABLE ACCESS FULL**). The column being searched must now be determined to see if an index is defined for that column. Looking again at the SQL query… By separating each clause of the SQL statement on a separate line it is easier to see the tables being accessed and the columns being searched in each table. Tables being searched appear after the **FROM** keyword. In this example, the columns being searched appear after the **WHERE** keyword:

```
SELECT DISTINCT t_02.PUID
   FROM PAPPEARANCEATTR t_01, PAPPEARANCE t_02
     WHERE ( ( t_02.rappearance_rootu = 'A180i4zyAAAMeC' )
       AND ( ( t_01.pvalue LIKE '3%'
       AND ( t_01.rdefinitionu = 'AN5w9yFrAAAMeC' ) )
       AND ( t_02.PUID = t_01.rappearanceu ) ) )
    INTERSECT SELECT t_04.PUID
    FROM PAPPEARANCEATTR t_03 , PAPPEARANCE t_04
       WHERE ( ( t_03.pvalue LIKE '4%'
AND ( t_03.rdefinitionu = 'AN8w9yFrAAAMeC' ) )
AND ( t_04.PUID = t_03.rappearanceu ) );
```

The example statement includes two **WHERE** clauses, each with three references to **`t_01`** (**`t_01`** is a table *alias* for **PAPPEARANCEATTR**[49]). Look for columns of the **PAPPEARANCEATTR** (**`t_01`**) table.

The columns in the **PAPPERANCEATTR** (or **t_0**1) table being searched are: **pvalue**[50], **rdefinitionu**, and **rappearanceu**.

In the table/index section, you notice that the indexes for the **PAPPEARANCEATTR** table are listed twice, because the table is accessed twice: once for each **WHERE** clause in the statement. The format of the index lines is:

```
index indexname on tablename <column1[, column2...) last
analyzed <date|never> over number rows
```

Looking at the PAPPEARANCEATTR table indexes:

```
table PAPPEARANCEATTR has 81744 rows, last analyzed 2002/07/ with
47514 rows
index PIPAPPEARANCEAT_2 on PAPPEARANCEATTR (PUID) last analyzed never
over 0 rows
index PIPAPPEARANCEAT_3 on PAPPEARANCEATTR (RAPPEARANCEU) last
analyzed never over 0 rows
index PIPAPPEARANCEAT_4 on PAPPEARANCEATTR (RAPPEARANCEU,
RDEFINITIONU) last analyzed never over 0 rows
```

[49] A table alias can be defined in an SQL query by including an alias name after the actual table name in the 'FROM' clause.

[50] When Teamcenter creates a property on an object, it creates a column for that property and prefixes the property name with p (designating a POM property). The *pvalue* column contains the Appearance Attribute property *value*.

The example shows three separate indexes for the **PAPPEARANCEATTR** table:

Index	Column
PIPAPPEARANCEAT_2	PUID
PIPAPPEARANCEAT_3	RAPPEARANCEU
PIPAPPEARANCEAT_4	RAPPEARANCEU and RDEFINITIONU

The **WHERE** clauses are searching the **pvalue** column (**t_01.pvalue LIKE …**) but there is no index for the **pvalue** column.

In Teamcenter the **pappearanceattr** table maps to the **appearanceattr** class and the **pvalue** column maps to the **value** property[51]. The **value** property is not normally indexed by the install/upgrade program, but since a custom query was created by the user that searches on the **value** property, an index should be added by the system administrator.

In general, you can remove the first **p** from the table and column reported in the explain plan for use as the class and property name respectively with the **–add_index** installation option.

To avoid the full table scan, an index must be defined on the **value** attribute of the **appearanceattr** class using **install –add_index** as follows:

```
$TC_BIN/install -add_index infodba <password> dba pvalue_index1 0
   appearanceattr value
```

Value	Description
PVALUE_INDEX1	Specifies the name of the new index (choose any name you like).
APPEARANCEATTR	Specifies the table.
VALUE	Specifies the property that needs an index.

The Teamcenter **install –add_index** function prepends the **p**, and creates an index on the **pvalue** column of the **pappearanceattr** table, and adds an entry to the Teamcenter POM data dictionary.

Once the missing index is properly created, this query should not take as long the next time the corresponding Teamcenter operation is executed. It should take just a fraction of

[51] The install program prefixes a ‘p’ (designating a POM property) to the property name and class name to create the column and table names respectively. So to add an index to the **pappearanceattr** table, use **appearanceattr** on the install command line. Similarly, use **value** on the command line to specify the **pvalue** column.

a second. For that reason you should not see this particular SQL in the syslog the next time it is executed with the **TC_SLOW_SQL** environment variable set.

NOTE

Remember, creating the index directly with SQL does not enter the index into the Teamcenter POM data dictionary. The **index_verifier** utility would then not be able to determine if the index ends up missing again. Wherever possible, use the **install – add_index** function to add indexes to Teamcenter.

Analyzing the Statistics: Oracle

The listing shows that tables were analyzed, but the number of rows has doubled since last time the analysis was done:

```
table PAPPEARANCE has 14493 rows, last analyzed 2002/07/10 with 7014
rows
```

The **PAPPEARANCE** table had 7,014 rows when it was analyzed on October 7th, 2002 (2002/07/10), but now has 14,493 rows. Out of date statistics may cause the optimizer to generate less than optimal execution plans.

The listing also shows the indexes involved were never analyzed:

```
index PIPAPPEARANCE_6 on PAPPEARANCE (RPARENTU) last analyzed
never over 0 rows
```

Always analyze indexes whenever tables are analyzed. Always use the **dbms_stats.gather_schema_stats** procedure to update both the table and index statistics at the same time

Oracle's DBMS_STATS.AUTO_SAMPLE_SIZE estimation parameter-value collects statistics faster and with nearly the accuracy of sampling 100% of the data.

You can use the DBMS_STATS.AUTO_SAMPLE_SIZE parameter-value to collect statistics provided there are no execution plan issues that result from its use. Databases with slow query executions (identified as slow query notes and/or execution plans in syslogs) should be analyzed at 100% to provide the best possible execution plan.

There are four basic commands that can be used to collect statistics.

5) Collecting statistics for an entire schema <u>where there is no expectation of problem query plans</u>:

```
exec dbms_stats.gather_schema_stats(ownname=>'infodba',
estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt=>'FOR ALL
COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true,
no_invalidate=>FALSE );
```

6) Collecting the best possible statistics for an entire schema:

```
exec dbms_stats.gather_schema_stats(ownname=>'infodba',
estimate_percent=>100, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true, no_invalidate=>FALSE );
```

NOTE:

You should reproduce performance issues using statistics collected at 100 percent before filing PRs associated with poor performance.

If you would like to collect statistics for a single table one of the following two commands can be used.

7) Collecting statistics on a single table where there is no expectation of problem query plans:

```
exec dbms_stats.gather_table_stats(ownname=>'infodba', tabname=>'<TABLE>',
estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt=>'FOR ALL
COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true,
no_invalidate=>FALSE );
```

8) Collecting the best possible statistics for a single table:

```
exec dbms_stats.gather_table_stats(ownname=>'infodba', tabname=>'<TABLE>',
estimate_percent=>100, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true, no_invalidate=>FALSE );
```

Just as with Function Based Indexes (FBIs), you can check for properties that may benefit from additional indexes by checking the database for recently executed SQL statements with high disk I/O. Use the SQL provided in section 5.2.3.3, *Function-Based Indexes* later in this chapter, and substitute **UPPER** in the **WHERE** clause with the property name (from the Schema Editor).

5.1.4 *Tablespace/Data File Organization*

Like configuration and maintenance, tablespace / data file organization is generally peculiar to the specific database management system being used. Refer to the sections later in this chapter for specific Oracle, SQL Server, or DB2 guidelines.

5.1.5 *Diagnosing Database Bottlenecks*

Assuming you have already optimized the server configuration and database parameters, the following approach should help identify database related Teamcenter bottlenecks.

Use Teamcenter to identify the queries that take the most time. Set the Teamcenter **TC_SQL_DEBUG** environment variable to **BPT** (for example, export **TC_SQL_DEBUG=BPT**). Remember to set **TC_KEEP_SYSTEM_LOG=Y**. Start a Teamcenter session, perform the slow operation (for example, a custom search), and exit Teamcenter.

Locate the syslog file generated for the session just executed. At the end of the syslog file, there is a listing of the most frequently performed SQLs, and statistics about how long they took. The result is similar to the example in Table 5-1. The statements are sorted by frequency; how many times they were executed during the session. There may be more than 200 statements listed; focus on the top dozen or so. **Cum.t**, **Max.t**, and **Avg.t** refer to cumulative, maximum, and average times to execute, respectively.

Table 5-1, Most Frequently Performed SQL Listing Sample

Line	Freq.	Cum.t	Max.t	Avg.t	Rows	Err.	Statement
1	63	0.560	0.491	0.009	63	0	`Select PAUDITDEFINITION.pauid, ptimestamp, ppid,rowning_siteu, rowning_sitec, pis_frozen,VLA_36_3, pdays_kept, rarchive_storageu, rarchive_storagec, pstorage_type, ruser_log_handleru, ruser_log_handlerc, pevent_type, pobject_type FROM PPOM_OBJECT, PAUDITDEFINITION WHERE PAUDITDEFINITION.puid = :1 AND PPOM_OBJECT.puid=PAUDITDEFINITION.puid`
2	60	0.044	0.005	0.001	0	0	`Select DISTINCT t_0.puid FROM PIMANCOMPOUNDPROPDEF t_0 WHERE (UPPER(t_0.pdisplay_type_name) = UPPER(:1))`
3	58	0.037	0.004	0.001	58	0	`Select ppid, puidfrom PPOM_OBJECT where puid IN (:1)`
4	56	0.045	0.008	0.001	50	0	`Select  DISTINCT t_01.rsecondary_objectu FROM PIMANRELATION t_01WHERE ( ( t_01.rprimary_objectu = :1 )AND ( t_01.rrelation_typeu = :2 )`

1. Search the list for statements that take excessive amounts of average time. In a LAN environment, most statements should execute in a few milliseconds[52].
2. Find the table names in the statement. One or more table names immediately follow the words **FROM**, **INTO**, or **UPDATE**. You can then research the database system to determine if the tables are properly indexed. Or you may decide that

[52] 1 millisecond = 0.001 second

these tables might benefit from being cached or located in a separate tablespace on another disk.

3. If Oracle is the database server, and a statement contains an SQL function (for example, **UPPER** or **SUBSTR**), the related table/column may benefit from a function-based index (FBI).
4. You can also set **TC_SLOW_SQL** to a time value to report SQL statements that take longer than that value, and then perform the slow operation. The advantage to this is that **TC_SLOW_SQL** includes an explain plan in addition to the slow queries.

NOTE

For more information on creating Oracle FBIs, see *Function-Based Indexes*, in the *Managing Oracle Database Servers* chapter.

5.2 Managing Oracle Database Servers

This section provides the necessary information to initially size and configure an Oracle database instance for Teamcenter.

NOTE

Teamcenter 8 and above does not support Oracle versions earlier than Oracle 10.2.0.1. If you use Oracle as your RDBMS, Teamcenter Environment Manager (TEM) verifies the Oracle version during installation. If the Oracle server does not meet the minimum required version, TEM does not install Teamcenter.

5.2.1 *Oracle Performance Tuning Quick Reference*

This section quickly helps validate Oracle configuration settings for Teamcenter. It assumes the reader has a reasonable understanding of the factors that affect performance and is familiar with terms defined later in this chapter. It is imperative the reader understands that tuning any database management system is an iterative process requiring careful record keeping and patience to measure, make configuration changes, and measure again, until optimal performance is achieved.

The tables below provide guidelines for initial configuration and performance tuning only when used with Teamcenter. Do not use this information until this section has been reviewed in its entirety.

Table 5-2 lists a number of well-known Oracle configuration settings and tuning methods that have the greatest impact to Teamcenter system performance. These should be checked to be certain they are correctly set or performed.

Table 5-2, Oracle Performance Tuning and Configuration Checklist

Description	Location / Notes
Be certain Cost Based Optimizer (CBO) Initialization Parameters are set: • OPTIMIZER_INDEX_CACHING = 95 • OPTIMIZER_INDEX_COST_ADJ = 10 • OPTIMIZER_ADAPTIVE_FEATURES=false (Oracle 12c only)	init<*sid*>.ora and/or pfile Change from previous releases See *Cost-Based Optimization*

Table 5-2, Oracle Performance Tuning and Configuration Checklist

Description	Location / Notes
Check memory allocation parameters <u>Auto Memory Management</u> • sga_target = *appropriate_size* • sga_max_size = *appropriate_size* • workarea_size_policy = AUTO • pga_aggregate_target = 20% * sga_target <u>Manual Memory Management</u> • db_cache_size = *appropriate_size* • large_pool_size = *appropriate_size* • shared_pool_size = *appropriate_size* • sort_area_size = 262,144 > 524,288 • java_pool_size = 0 <u>Either</u> • db_block_size = 8192 • db_file_multiblock_read_count = 16 • undo_management = AUTO	init<*sid*>.ora and/or pfile **Measurements have shown Tc 8.3 and later requires increased SGA / PGA allocations compared with previous Tc versions.** For *appropriate_size* see *System / Program Global Area (SGA / PGA)*. Remember, these are suggested starting values only.
Routinely update the CBO table and index statistics • `exec dbms_stats.gather_schema_stats(ownname=>'infodba', estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt=>'FOR ALL COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true, no_invalidate=>FALSE );`	See other options in *Update Table and Index Statistics* Any other statistics gathering procedures, including jobs scheduled in a default Oracle installation should be modified to conform to this sample.
Check OS Kernel Parameters affecting memory utilization AIX: • mkdev -l aio0 • smit aio Solaris: • set fastscan = 131072 • set maxpgio = 65536	See *Operating System Considerations* Remember, these are suggested starting values only.

Table 5-2, Oracle Performance Tuning and Configuration Checklist

Description	Location / Notes
Check for high disk I/O, especially against single tables	See *Determining Frequently Accessed Tables*
Check that Indexes were created for Saved Queries and Schema Extensions	See *Saved Queries* and *Extended Data Model Indexes*
Disable slow SQL reporting except to diagnose performance issues: • TC_SLOW_SQL=-1	See *Slow SQL Reporting*

5.2.2 ***Oracle Server Performance***

This section provides information generally applicable to most customer deployments using an Oracle database. Sample SQL statements are provided that can be used to monitor and tune some areas of the Oracle system for the novice system administrator. Experienced Oracle database administrators likely have techniques of their own.

5.2.2.1 Oracle Enterprise Manager

The Oracle Enterprise Manager (OEM) is highly recommended to manage and tune Oracle performance for Teamcenter. OEM provides general management and maintenance capabilities as well as performance graphs and tools to assess workload, server utilization, top database activity, and performance alerts.

NOTE

Not all OEM features are available in all Oracle license editions. At the time of this writing, *Database Control, automatic configuration* is included in Standard and Enterprise Editions (SE1, SE, and EE). Consult the Oracle documentation for additional information about the Oracle Enterprise Manager.

An example of the *OEM Console* is shown in Figure 5-2, which provides a graphical, intuitive interface to manage and monitor basic Oracle functions on any server or instance in your enterprise. Several automated features may be available, or available with additional licensing fees including:

Automated Workload Repository	Automatically collects performance metrics for reporting or comparison at a later time
Automatic Database Diagnostic Monitor	Monitors key metrics and generates alerts with configuration recommendations to improve performance
SQL Tuning Advisor	Provides recommendations to tune SQL queries to improve performance or reduce resource requirements

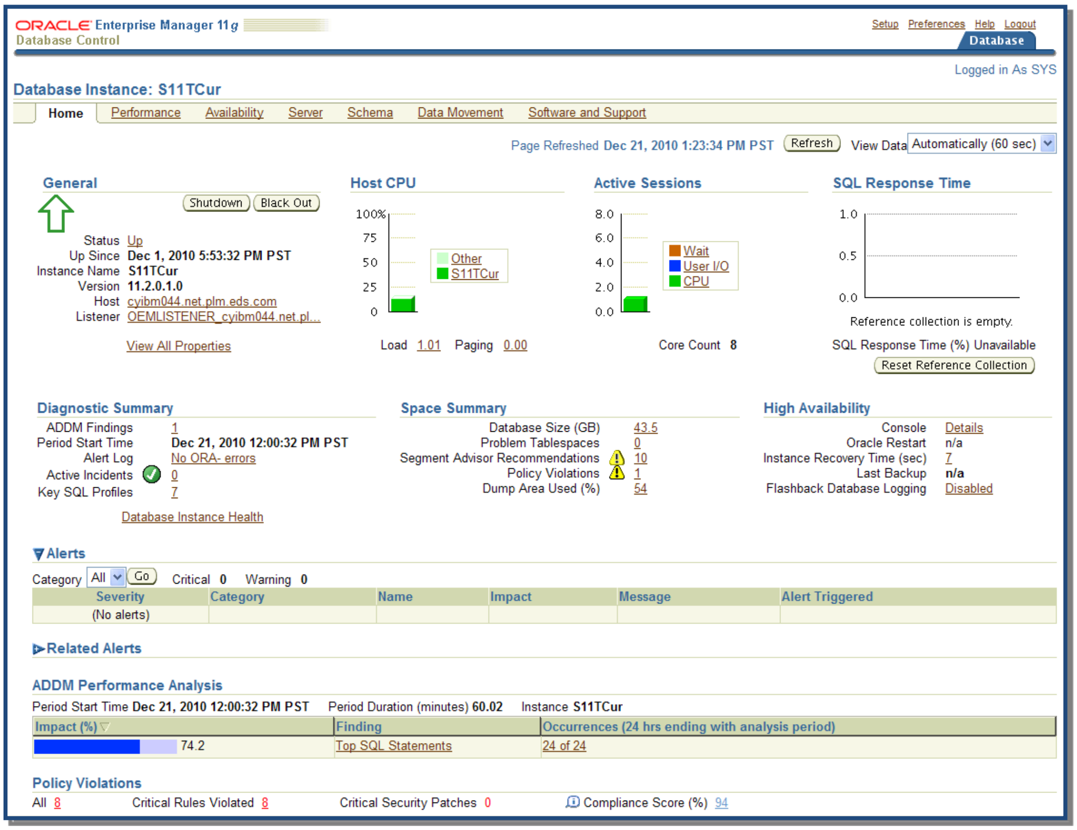


Figure 5-2, Oracle Enterprise Manager

Set aside some time to become familiar with how to navigate and interpret the OEM Console.

5.2.2.2 Teamcenter Oracle Performance Issues

Operating System Considerations

Oracle requires a number of operating system kernel parameters be adjusted for optimal operation on a UNIX server. See also the *Teamcenter Help Collection - Installing - Installation on UNIX and Linux Servers Guide - Database Server Installation*.

In addition to the required settings, the following kernel parameters may have an impact on Oracle performance:

- Operating System Buffer Cache

- Memory page scan rates/algorithms
- Server I/O settings

AIX

Oracle performs better with asynchronous I/O enabled. On AIX 6 and AIX 7, the Asynchronous Input Output (AIO) device drivers are enabled by default. See also http://docs.oracle.com/database/121/AXDQI/toc.htm#CDEGIBFD.

If by some chance you still have AIX 5L or earlier you should enable it as follows:

Log in as root at shell prompt.

Enter the following command to verify whether asynchronous I/O services are running. If they are not, the command will start the services:

```
# mkdev -l aio0
```

To ensure the services are configured with each system restart:

At the command prompt, enter the following:

```
# smit aio
```

Select the menu item: Change/Show Characteristics of Asynchronous I/O.

Change the field: "STATE to be configured at system restart" from DEFINED to AVAILABLE.

Solaris 10

See also http://docs.oracle.com/database/121/AXDQI/toc.htm#CDEGIBFD.

By default, Solaris limits the amount of shared memory available to a single user to a percentage of total physical memory. If you want to set a very large SGA, or run multiple instances that exceed this limit, you will need to increase the available shared memory with Solaris 10 Resource Management. For example the following will increase shared memory to 6 GB and set `max-sem-nsems` to 2048 for the Oracle user.

```
projadd -c "Oracle Project" -U oracle,root -K "project.max-shm-memory=(priv,6GB,deny)" -K "process.max-sem-nsems=(priv,2048,deny)" user.oracle
```

For more information about Solaris shared memory see http://docs.oracle.com/database/121/UNXAR/appe_sol.htm#UNXAR013.

Windows

None required at the time of this writing.

Suse

None required at the time of this writing.

NOTE

Actual kernel parameter names vary by operating system and version; consult your platform-specific system administration guides. For other UNIX systems, there may be equivalent or similar parameters. For these systems the parameters and values listed here can only be used as a guide for where to consider changes.

5.2.3 *Oracle Indexes*

5.2.3.1 Verifying Oracle Indexes Manually

If you created indexes with SQL, or suspect one or more table indexes are missing, you can verify their existence within OEM or with the following SQL statements:

```
select index_name, substr(column_name, 1, 32), column_position
  from user_ind_columns
  where table_name = '<tablename>'
  order by index_name, column_position;
```

The output should be similar to:

```
INDEX_NAME SUBSTR (COLUMN_NAME, 1, 32)
-------------- -------------------------------
PIPPOM_STUB  PUID
PIPPOM_STUB_0  POBJECT_UID
PIPPOM_STUB_2  SYS_NC00006$;
```

To see what indexes are defined against which columns, you can also use the following SQL:

```
select index_name, substr(column_name, 1, 32), column_position
  from user_ind_columns
  where table_name = '<tablename>'
  order by index_name, column_position;
```

For example, checking the indexes for the **PAPPEARANCEATTR** table as above should then result in something like the following:

```
INDEX_NAME SUBSTR(COLUMN_NAME,1,32)COLUMN_POSITION
-------------------- -------------------------------- ---------------
PIPAPPEARANCEAT_2    PUID       1
```

```
PIPAPPEARANCEAT_3    RAPPEARANCEU        1
PIPAPPEARANCEAT_4    RAPPEARANCEU        1
PIPAPPEARANCEAT_4    RDEFINITIONU        2
PVALUE_INDEX1        PVALUE    1
```

If the column name returned is similar to **SYS_NC00006$** as in this example, this is an FBI index. You can examine the FBI index expression with:

```
select index_name, column_expresssion, column_position
  from user_ind_expressions
  where table_name = '<tablename>';
```

This might produce something like:

```
INDEX_NAME COLUMN_EXPRESSION
------------- ----------------------------
PIPPOM_STUB_2  SUBSTR("POBJECT_UID",1,14)
```

In this example, the **PIPPOM_STUB_2** index is a function-based index (FBI) containing the first 14 characters of the **POBJECT_UID** column.

Once you verify that the required indexes exist, consider archiving a list of them for comparison at a later date. To get a complete list of indexes, use the following SQL:

```
set pagesize 1000
spool Teamcenter_indices_<date>
select table_name, index_name, uniqueness
  from user_indexes order by 1, 2;
select table_name, table_name, column_name, column_position
  from user_ind_columns order by 1, 2, 3, 4;
spool off
```

The first query gives the names and whether an index is unique or not. The second gives the columns and the order they appear in the index definition. If there are questions about missing indexes unknown to the Teamcenter data dictionary, you can list them all again and compare it with the archived version.

Alternatively verify indexes in OEM under *Administration(tab)→Schema→Database Objects→Indexes*.

A more complete set of index information can be obtained using Oracle's DBMS_METADATA package. The following script will generate the complete SQL (indicating uniqueness, storage options, etc.) for all indexes in the Teamcenter schema. (Note that it may take a couple of minutes to execute.)

```
set pages 0
set lines 120
set trimspool on
set long 1000000
set longchunksize 1000000
spool teamcenter_indexes_ddl_<date>.sql
select dbms_metadata.get_ddl('INDEX',index_name)||'/'
  FROM user_indexes ORDER BY table_name, index_name;
spool off
```

Having this script makes it possible to restore indexes quickly simply by cutting and pasting the specific `CREATE INDEX` statement into SQLPlus.

5.2.3.2 Index Maintenance

It is very important that indexes are monitored and maintained routinely to ensure continued performance, especially when using Oracle's Cost-Based Optimization (CBO) as recommended by Siemens PLM Software. As information is added to or modified in the Teamcenter system, statistics can go out of date or the indexes can become excessively deep.

Run the Index Verifier Routinely

Occasionally run the **index_verifier** utility as described earlier in this chapter to be certain indexes were not inadvertently dropped. Although uncommon, accidents do happen. It is not impossible for a novice administrator to remove an important index inadvertently.

Update Table and Index Statistics

Always analyze indexes whenever tables are analyzed. Always use the **dbms_stats.gather_schema_stats** procedure to update both the table and index statistics at the same time

Oracle's DBMS_STATS.AUTO_SAMPLE_SIZE estimation parameter-value collects statistics faster and with nearly the accuracy of sampling 100% of the data.

You can use the DBMS_STATS.AUTO_SAMPLE_SIZE parameter-value to collect statistics provided there are no execution plan issues that result from its use. Databases with slow query executions (identified as slow query notes and/or execution plans in syslogs) should be analyzed at 100% to provide the best possible execution plan.

There are four basic commands that can be used to collect statistics.

9) Collecting statistics for an entire schema <u>where there is no expectation of problem query plans</u>:

```
exec dbms_stats.gather_schema_stats(ownname=>'infodba',
estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt=>'FOR ALL
```

```
COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true,
no_invalidate=>FALSE );
```

10) Collecting the best possible statistics for an entire schema:

```
exec dbms_stats.gather_schema_stats(ownname=>'infodba',
estimate_percent=>100, method_opt=>'FOR ALL COLUMNS SIZE AUTO',
degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true, no_invalidate=>FALSE );
```

NOTE:

You should reproduce performance issues using statistics collected at 100 percent before filing PRs associated with poor performance.

If you would like to collect statistics for a single table one of the following two commands can be used.

11) Collecting statistics on a single table where there is no expectation of problem query plans:

```
exec dbms_stats.gather_table_stats(ownname=>'infodba', tabname=>'<TABLE>',
estimate_percent=>DBMS_STATS.AUTO_SAMPLE_SIZE, method_opt=>'FOR ALL
COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true,
no_invalidate=>FALSE );
```

12) Collecting the best possible statistics for a single table:

```
exec dbms_stats.gather_table_stats(ownname=>'infodba',
tabname=>'<TABLE>',  estimate_percent=>100, method_opt=>'FOR ALL
COLUMNS SIZE AUTO', degree=>DBMS_STATS.AUTO_DEGREE, cascade=>true,
no_invalidate=>FALSE );
```

For more information on scheduling Oracle jobs, see the Oracle *Administrator's Guide* (on the Oracle documentation CD-ROM or online at oracle.com) for details on using DBMS_JOB or DBMS_SCHEDULER. You can also schedule jobs with the OEM console; see the relevant chapter in the *Oracle Enterprise Manager Concepts* guide. How often the CBO statistics should be updated depends on how much the contents of the database change over time.

If users frequently create or update information, or import large sets of data in a short period of time, run the **dbms_stats.gather_schema_stats** procedure often; the faster they create or update information, the more frequently it should be run. At an absolute minimum, statistics should be gathered weekly, although a daily run is far preferable. Oracle's job scheduling mechanisms will allow this to be scheduled for a time of day with low-usage, and (via resource plans) in such a way that avoids adverse system performance during the gathering. (Consult the Oracle documentation for more information on these features) If massive data imports are being performed on a production system, consider running this even more often.

Manually Creating and Analyzing Indexes

It is best to create indexes with the **install –add index** utility. If you create indexes manually using SQL, the index is not entered into the Teamcenter POM data dictionary. The index_verifier utility would then not be able to determine if the index ends up missing again.

Under extraordinary circumstances, an index can become corrupt. If this is suspected, you can manually remove and re-create the index. First, be sure to find the tablespace that you use to keep indexes, either with OEM under *Administration(tab)→Schema→Database Objects→Indexes* or with the following SQL query:

```
select distinct tablespace_name
     from user_indexes
     where table_name='<tablename>';
```

Now drop the suspect index and re-create it in that tablespace with the tablespace name returned by the above query:

```
DROP INDEX index_name
CREATE [UNIQUE] INDEX index_name ON tablename ("column name"…)
TABLESPACE tablespace;
```

Alternatively, retrieve the precise syntax of the index using

```
set pages 0
set longchunksize 1000000
set long 1000000
set lines 120
SELECT dbms_metadata.get_ddl('INDEX', '<index_name>') FROM dual;
```

Then, issue:

```
DROP INDEX <index_name>;
```

Then enter the create command that was returned from the SELECT statement above (using cut and paste).

After creating the index, be certain to analyze it. Analyze index and table statistics with the **dbms_stats.gather_schema_stats** procedure (as described previously) or use *OEM→Administration→Statistics Management→Manage Optimizer Statistics*.

5.2.3.3 Function-Based Indexes

To improve performance of a number of Teamcenter queries, Teamcenter uses Function-Based Indexes (FBI) for some tables. FBIs have been shown to improve Oracle performance for a number of customers with large databases that use case-insensitive searches or other functions. The installation and upgrade procedures automatically create Oracle FBIs for operations that have been shown to benefit from them.

In addition to the FBIs created at installation and upgrade, additional FBIs may improve performance if the Teamcenter data schema has been extended. As always, consult the Oracle system administration documentation for complete details.

You can check for other properties that FBIs may benefit by checking the database for recently executed SQL statements using functions with high disk I/O. If a query appears in the **v$sqlarea** dynamic system view as high in disk I/O and uses a function such as **UPPER** or **SUBSTR** on some table/column, it should be regarded as a candidate for an FBI. Use the following SQL to list high I/O statements:

```
set pagesize 1000
spool /tmp/high_diskio_sql.txt

select DISK_READS, SQL_TEXT
  from v$sqlarea
  where DISK_READS > 0
  and SQL_TEXT like ('%UPPER%')
  order by DISK_READS desc;

spool off
```

The result can return thousands of rows. Adjust the value (0) of the "**where DISK READS > 0**" clause as desired to get a manageable count, then decrease the value incrementally to find more opportunities for FBIs. If a corresponding FBI does not exist, create it as detailed above. To search for other functions, replace **UPPER** with the desired function name. Alternatively, you can use the Statspack or OEM Console to find statements with high disk I/O.

To create FBIs on any attribute for which functions are used, execute the following SQL statement:

```
CREATE [UNIQUE] INDEX index_name ON table-name (FUNCTION (column-name))
   [TABLESPACE <tablespace_name>];
```

For example, to create an FBIs for the **UPPER** function, use the following SQL statement:

```
CREATE INDEX pobject_name_upper_ind on
   pworkspaceobject(upper(pobject_name);
```

NOTES

UPPER is an example; most functions supported by Oracle RDBMS can be used.

After creating new FBIs, be certain to analyze table statistics for all tables in the infodba schema with the **dbms_stats.gather_schema_stats** procedure as documented in *Index Maintenance*, later in this chapter.

5.2.3.4 Separate Index Tablespace

Contrary to previous Siemens PLM Software recommendations, recent evidence with Oracle suggests that moving indexes into separate tablespace does not significantly improve query performance, and in some cases may degrade performance somewhat. Keeping indexes with their respective tables has been shown to have better results. If a particular tablespace has a significant amount of disk I/O, consider splitting out heavily used tables, along with their indexes, as described in *Segregating Tables*, later in this chapter.

5.2.3.5 Adding New Column in pom_backpointer Index

The **pipom_backpointer2** index defined on the **to_uid** column can be replaced with an index defined on the **to_uid** and **from_uid** columns on the **pom_backpointer** table to improve the performance and also to reduce the input/output cost of the query, for example:

```
select to_uid from pom_backpointer where from_uid='value';
```

This may degrade the performance of other queries if the SGA is not sized properly, since the blocks cached in SGA will be at the expense of other blocks. So without appropriate SGA available, adding an index to a default setup could reduce the performance in other areas (induce input/output operations for other queries). The index can be tested to see if the performance gains are without downsides.

Please refer section 5.2.4.1 System / Program Global Area (SGA / PGA) to resize the SGA.

SQL statements to drop/create index:

```
Drop index pipom_backpointer2 on pom_backpointer;

Create unique index pipom_backpointer3 on
   pom_backpointer(to_uid,from_uid) tablespace idata;
```

5.2.4 Oracle Initialization Parameters

The default Oracle initialization parameters supplied with Teamcenter are appropriate for typical customer deployments (100–250 users). Small, medium, and large scenarios included in the default files cover deployments of 50, 100, and 200 users, respectively. For larger deployments, some parameters require adjustment, and some require adjustment depending on actual usage patterns. Others can only be properly set once statistics are gathered in a production environment. In general, start with the default settings unless your deployment:

- Serves more than 250 users
- Is predominately used by casual users

If either of these situations applies, consider the following changes to the Oracle initialization parameters (stored in **init.ora** or in the database spfile). If using init.ora, be sure to back up the original file first. Changes should be made one at a time, with monitoring and tracking of system function and performance after each change.

5.2.4.1 System / Program Global Area (SGA / PGA)

For more than 250 users, the size of the System Global Area (SGA) and Program Global Area (PGA) will need to be increased to support the larger number of users. Measurements with previous versions of Teamcenter have shown that deployments of nearly 1,500 users can require an SGA up to 4 GB in size. Monitor SGA / PGA utilization with OEM and adjust the size as needed.

Also adding new indexes will have an impact on the System Global Area, so the SGA should be monitored after either the new indexes are added or the product has been upgraded.

NOTE:

Nearly every major version of Teamcenter leverages more database features that often require somewhat more SGA / PGA memory. Measurements with Tc 11.2.1 have shown additional SGA / PGA does benefit some operations compared to previous Teamcenter versions. Always, monitor SGA / PGA utilization with OEM after upgrading Teamcenter and adjust the size as needed for your production deployment.

System Global Area (SGA)

In Oracle 10*g* or later, the sizes of the various SGA components (as well as user processes[53]) can be automatically (dynamically) adjusted by Oracle’s Automated Shared Memory Management (ASSM) feature. You can continue to manually set the individual components of SGA by setting the parameters db_cache_size, java_pool_size, large_pool_size, shared_pool_size, however Oracle generally recommends moving to ASSM dynamic SGA allocations for most deployments. Dynamic SGA management is enabled by setting values for **sga_target** and setting **statistics_level** to **TYPICAL** or **ALL**.

[53] Program Global Area (PGA)

NOTE

Some customers with very large SGAs (≥ 4 GB) have experienced poor performance with a small set of Oracle operations when using ASSM. This typically happens if part of the buffer cache is being dynamically shrunk and a session wants access to a data block that resides in a db cache buffer within a granule chosen to be freed. If you experience performance issues when using ASSM, validate this cause by reverting to static SGA management (remove or disable (by setting it to 0) the **sga_target** parameter, and set the conventional parameters for SGA (i.e. db_cache_size, shared pool, java pool, large pool)), and consult your Oracle support team.

Always monitor SGA utilization and increase the size of the SGA with **sga_target** and **sga_max_size** if warranted. **sga_target** can be increased or decreased at any time; **sga_max_size** is set at instance startup and sets the largest value to which **sga_target** can be set. If using Oracle 10*g* with more than 250 users, consider the following initial SGA sizes:

Number of Users	SGA Size
250–500	**sga_target =** 512 MB–1.0 GB, **sga_max_size** = 768 MB–1.5 GB)
500–1000	**sga_target =** 1.5 GB –3.0 GB, **sga_max_size** = 2.0 GB –4.0 GB)
> 1000	**sga_target =** 3.0GB or larger, **sga_max_size** = 4.0GB or larger)

NOTE

It is not productive to set SGA larger than physical memory; monitor system memory page rates. It may be necessary to install more physical memory.

If you have limited memory and want to maximize SGA size, use the following approach with Oracle 10*g* or later:

- Subtract from total physical memory the operating system requirements (including a small operating system buffer cache)
- Allocate 80% of the remainder to SGA (sga_max_size)
- Allocate the final portion to PGA (pga_aggregate_target)

Once initial SGA values have been set, frequently assess SGA utilization with OEM/AWR during production use, modify the configuration incrementally, and monitor regularly to ensure optimal configuration.

The *Buffer Pool Advisory* table in the AWR report can help determine if more memory should be allocation to SGA, or if less can suffice; an example is shown in Table 5-3 below. The row with Size Factor = 1.00 ① shows the current setting and resulting buffer activity. Higher size factors estimate buffer activity at incrementally larger buffer sizes. The example in Table 5-3 shows that at the current SGA setting, 816 MB is allocated to the database buffer pool causing 3,986,650 physical reads. At nearly double the size (i.e.

1.96 x ②) Oracle estimates physical reads would be reduced only 8% to approximately 3,651,931. With this information the DBA can balance memory requirements with potential performance affects. Note the AWR report include advisories on a number of SGA pool structures.

Table 5-3, Example Oracle Buffer Pool Advisory

P	Size for Est. (M)	Size Factor	Buffers for Estimate	Est. Phys Read Factor	Estimated Physical Reads	
D	80	0.10	9,895	4.28	17,054,386	
D	160	0.20	19,790	2.69	10,736,254	
D	800	0.98	98,950	1.00	3,996,003	
D	816	1.00	100,929	1.00	3,986,650	1
D	880	1.08	108,845	0.99	3,939,680	
D	1,360	1.67	168,215	0.93	3,698,085	
D	1,440	1.76	178,110	0.92	3,677,339	
D	1,520	1.86	188,005	0.92	3,660,322	
D	1,600	1.96	197,900	0.92	3,651,931	2

Remember that "bigger is not always better." As already stated, it is *very* important that your SGA size, plus the other memory needed to run your workload, is less than the physical memory of the system so that paging does not occur. However, the notion that more memory is always better is not always the case with Oracle[54]. If you increase the size of SGA and no performance (or hit ratio) improvement is noticed, reduce the SGA to a lower value, or back to what it was before the change. Also, if the SGA hit ratio is high but you suspect the SGA may be too large, monitor LRU wait states to determine if the instance is spending excessive time managing SGA memory.

Program Global Area (PGA)

Beginning with 9*i*, Oracle automatically and dynamically adjusts the above parameters.

The **pga_aggregate_target** and **workarea_size_policy** initialization parameters control the total size of memory allocated to all user processes. To enable auto PGA management, set **workarea_size_policy** to AUTO and **pga_aggregate_target** according

[54] It has been noted on some OLTP workloads (similar to Teamcenter) that bigger SGAs can actually degrade performance even without causing paging/swapping. Possibly the overhead caused by a large number of pages (and increasing sizes of LRU lists) is actually greater than what it takes to perform the required I/O to read database blocks. This is not well understood, so always measure the effect of each change.

to the guidelines above for SGA. Automatic PGA memory management is the recommended setting for Teamcenter.

As a starting point, set **pga_aggregate_target** to 20% of SGA, then monitor PGA utilization with OEM and adjust the size as needed

Shared Pool Size

If auto SGA management is used with 10*g*, the value **shared_pool_size** is not used.

Processes

Set the number of processes slightly above the number of users that are expected to be logged in concurrently, plus 12 (to cover Oracle processes), plus a few more for administration purposes.

Cost-Based Optimization

There are several options for the cost-based optimizer (CBO) used in Oracle 10g forward. To support the CBO fully, both table and index statistics must be maintained regularly, in accordance with optimizer guidance given elsewhere in this chapter.

Teamcenter recommendations for the following initialization parameters have changed and should be set as indicated:

- OPTIMIZER_INDEX_CACHING = 95
- OPTIMIZER_INDEX_COST_ADJ = 10
- OPTIMIZER_ADAPTIVE_FEATURES=false (Oracle 12c only)

OPTIMIZER_MODE

Previous deployment guidelines recommended setting **OPTIMIZER_MODE = ALL_ROWS**. This is no longer required with Oracle 10g or later

OPTIMIZER_INDEX_CACHING & OPTIMIZER_INDEX_COST_ADJ

Previous deployment guidelines were to unset OPTIMIZER_INDEX_CACHING and OPTIMIZER_INDEX_COST_ADJ according to the latest recommendation from Oracle for 10g and 11g. Recent field experience since those guidelines were published suggests setting OPTIMIZER_INDEX_CACHING to 95 and OPTIMIZER_INDEX_COST_ADJ to 10 provide better performance for Teamcenter in general.

It is possible the default values will not produce optimal performance. Gathering statistics during production use creates ‘workload’ system statistics, which should provide better performance results in most cases. Workload system statistics should be gathered during a period when the server is under normal load. It is better to do this when the system is live, vs. a simulation, as synthetic benchmarks do not truly reflect real load.

Consider collecting them for one full work shift or day. To collect workload system statistics for a specific period of time execute the following SQL statements.

To start gathering statistics at the beginning of the work day

```
dbms_stats.gather_system_stats(gathering_mode => 'start')
```

To stop gathering statistics at the end of the work day

```
dbms_stats.gather_system_stats(gathering_mode => 'stop')
```

Alternately, you can set Oracle to collect system statistics continuously at fixed intervals.

```
dbms_stats.gather_system_stats(gathering_mode => 'interval', interval => 30)
```

However, once gathered, there is generally no need to recalculate the system statistics unless there is significant change to the workload, server hardware, or Oracle configuration.

As always, measure the performance impact with every change to the Oracle configuration.

Use the settings that provide the best performance for your deployment.

5.2.5 *Oracle Table Management and Tablespace Organization*

If the Oracle server hardware and operating system have been optimized, and the Oracle configuration and indexes are tuned and managed, additional performance improvement may be possible with careful management of tables and tablespaces.

Some tables are accessed so frequently by so many users; it is worthwhile to *cache* them permanently. This configures Oracle to retain blocks from these 'hot' tables in memory (the SGA buffer cache) once they have been loaded, eliminating all disk access to read these blocks and improving table access times. Keeping tables permanently cached in SGA is often referred to as *pinning* them in memory. Database blocks of a cached table are brought into memory when needed but are then retained in a special cache. When they are updated, the new data is written to disk. This can improve performance as long as the cached tables are not so large that other tables contend for the remaining space in the SGA. The SGA buffer cache may become overcrowded if there are too many cached tables.

Large tables that are frequently accessed can be reorganized into separate tablespaces and put on separate disks or disk volumes. Doing so places these 'hot' tables on different disk paths, which can reduce contention for certain disk drives and again improve table access times. Reorganizing hot tables into separate tablespaces is not suggested for the novice

Teamcenter system administrator; however the general concepts are presented in this chapter. Consult an experienced Oracle database administrator for assistance.

In general, cache small hot tables (<100 rows) and segregate large hot tables into separate tablespaces rather than pinning them. Caching tables is a bit more straight-forward than reorganizing tablespaces, so the former will is addressed first.

For detailed information on these strategies, see the appropriate Oracle system administration documentation.

5.2.5.1 Determining Frequently Accessed Tables

Improving table access times requires two steps: determining the hot tables, and altering the instance to cache or segregate them.

You can get a good idea of what other tables are frequently accessed with the following SQL:

```
select DISK_READS, SQL_TEXT from v$sqlarea order by disk_reads;
```

This returns a list of SQL statements with the corresponding disk reads, sorted ascending by disk reads. Looking at the statement with the most disk reads first, find the table names in the statement. One or more table names immediately follow the words **FROM**, **INTO**, or **UPDATE**. These tables may be candidates for caching if they are small, or for their own tablespace if they are large.

Determine the size of a table with:

```
select count(*) from tablename;
```

If the count returned is more than a few hundred, it may be inefficient to cache it into memory.

5.2.5.2 Pinning – Permanently Caching Frequently Used Tables

Experience at a number of customer sites has shown that caching the following tables can be beneficial:

- PAM_ACE
- PAM_ACL
- PATTACHMENT_TYPES
- PATTACHMENTS
- PEPMTASK
- PIMANTYPE
- PITEMMASTER

- PITEMVERSIONMASTER
- POM_F_LOCK
- POM_M_LOCK
- POM_R_LOCK
- PPOM_USER
- PPSVIEWTYPE
- PSIGNOFF
- PUSER

These tables are small but are accessed continuously, so caching them almost always improves performance somewhat.

To cache a table into memory, a ‘keep’ buffer pool must first be created by adding an entry to the init<*sid*>.ora file:

`db_keep_cache_size = (`*`buffers:number`*`)` (10g or later)

Determine the *size* required for the keep buffer pool by summing the blocks for the tables to be cached. For example:

```
select sum(blocks)
 from dba_tables
 where table_name in(
  'PAM_ACE',
  'PAM_ACL',
  'PATTACHMENT_TYPES',
  'PATTACHMENTS',
  'PEPMTASK',
  'PIMANTYPE',
  'PITEMMASTER',
  'PITEMVERSIONMASTER',
  'POM_F_LOCK',
  'POM_M_LOCK',
  'POM_R_LOCK',
  'PPOM_USER',
  'PPSVIEWTYPE',
  'PSIGNOFF',
  'PUSER'
  );
```

This will return the number of blocks similar to:

```
SUM(BLOCKS)
-----------
        231
```

Then multiple the blocks used by 4 or 5 for potential growth and round up to the nearest hundred.

Next, cache the tables into memory:

```
alter table tablename storage(buffer_pool keep);
```

For example:

```
alter table POM_M_LOCK storage(buffer_pool keep);
```

To remove tables from the keep pool:

```
alter table tablename storage(buffer_pool default);
```

NOTE

If a statement listed from

```
select DISK_READS, SQL_TEXT from v$sqlarea order by disk_reads;
```

contains a *where* clause with an SQL function (for example, **UPPER** or **SUBSTR**) the table containing the referenced column might benefit from a function-based index (FBI) as well.

Segregating Tables

The list of SQL statements with hot tables probably contains tables, such as **POM_BACKPOINTER** or **PPOM_OBJECT**, that are also accessed very frequently. However, the size of these usually precludes caching them. Large tables should not be cached.

Frequently accessed large tables may be candidates for their own tablespace on a separate disk, however, you might choose to group tables into tablespaces. For example:

- Place the top three tables into their own separate tablespaces
- Place the next two tables into a separate tablespace
- Place the next five tables into a separate tablespace
- Place the remaining tables into a separate tablespace

This would result in a total of six tablespaces. As you segregate tables into separate tablespaces, recheck for hot tables and further segregate them as needed to achieve balance and optimize performance. Remember to keep indexes in tablespace with their respective table.

Commonly Segregated Tables

As stated, reorganizing hot tables into separate tablespaces is not suggested for the novice Teamcenter system administrator. However, experience with a number of customers shows that organizing the following tables is separate tablespaces can be beneficial, but only if each tablespace is on a separate disk drive or file system volume. Consider the following tablespace configurations with the assistance of an experienced Oracle DBA:

- Place the **POM_BACKPOINTER** table and its indexes in its own tablespace.
- Place the **PPOM_OBJECT** table and indexes in its own tablespace.
- Place the **POM_M_LOCK, POM_R_LOCK, POM_F_LOCK** tables and indexes in one tablespace.
- Place the **PEPMTASK** table and its indexes in one tablespace with its associated VLAs attachments and attachment types.

For example, to move the **POM_BACKPOINTER** table and its indexes to its own tablespace:

```
ALTER TABLE POM_BACKPOINTER MOVE TABLESPACE IDATA2;
ALTER INDEX PIPOM_BACKPOINTER REBUILD TABLESPACE IDATA2;
ALTER INDEX PIPOM_BACKPOINTER2 REBUILD TABLESPACE IDATA2;
```

Use the techniques defined above to identify hot tables for other candidate tables to be segregated using:

```
select DISK_READS, SQL_TEXT from v$sqlarea order by disk_reads;
```

5.2.6 *Finding Oracle System Bottlenecks*

Finding Oracle system bottlenecks can be a challenging task. Experience and patience improve your ability to determine if an Oracle configuration issue is the source of specific Teamcenter performance problems.

5.2.6.1 Monitor Oracle Statistics

Institute a process to regularly capture key Oracle statistics and monitor them for changes from previous measurements. This helps identify performance trends that you can correct, hopefully before users notice an impact. Standard Oracle tools, like Statspack or OEM provide information on the key metrics to monitor.

Oracle Enterprise Manager

The Oracle Enterprise Manager (OEM) is highly recommended to monitor and correct Oracle related performance issues. Refer to section 5.2.2.1 above for additional information about OEM.

Installing Oracle Statspack

Siemens PLM Software recommends the use of the Oracle Enterprise Manager (OEM) rather than Oracle Statspack.

Oracle Performance Statistics

There are a number of *virtual* tables maintained by Oracle that store performance related statistics. Table 5-4 lists some of the performance views typically available in Oracle. It is recommended that you become familiar with these tables.

Table 5-4, Sample Oracle Performance Views

View	Contents
V$SGA	Contains summary information on the system global area (SGA).
V$SGASTAT	Contains detailed information on the system global area (SGA)
V$SHARED_POOL_RESERVED	This fixed view lists statistics that help tune the reserved pool and space within the shared pool.
V$DBFILE	Contains data file information from the control file.
V$BUFFER_POOL_STATISTICS	Displays information about all buffer pools available for the instance
V$FILESTAT	Contains information about file read/write statistics.
V$LATCH	Lists statistics for non-parent latches and summary statistics for parent latches.
V$LIBRARYCACHE	Contains statistics about library cache performance and activity.
V$ROLLSTAT	Contains rollback segment statistics.
V$ROWCACHE	Contains statistics about data dictionary cache performance and activity.

Alternatively, most of these statistics are available in the OEM Console, which provides a more graphical, user friendly view, as well as a number of performance alerts and tuning advice. Refer to section 5.2.2.1 above for additional information about OEM. You can also enable Statspack to generate a baseline and then gather performance statistics hourly, daily, or weekly to track performance trends. For Dynamic Performance I/O views and Performance Tuning with Statspack, see the Oracle documentation.

Routinely check the Oracle statistics for tables that exhibit high levels of I/O. What is *high* is dependent on how your business uses Teamcenter, so monitor table space I/O in the File IO Stats section of the statspack report. Some regular checks of SQL by I/O can be helpful to establish what is normal for your system. Use OEM AWR, or get a list of the top queries by disk I/O with the following query:

```
select DISK_READS, SQL_TEXT from v$sqlarea order by disk_reads
```

Again, find the table names in the statements. These tables may also be candidates for their own tablespace on a separate disk.

5.2.6.2 Oracle File System Optimization

Storage subsystem performance is a very important aspect of tuning an Oracle database server for optimal performance. Read and write performance of the file system affects Oracle as follows:

Random read performance

- Important for Oracle indexed or hash-based queries and rollback segment reads, which are heavily used in Teamcenter.

Random write performance:

- Important for Oracle DBWR process writes, also heavily used in Teamcenter.

Sequential read performance:

- Backups, Oracle full table scans, index creations, temporary segment reads, and recovery from archived redo log files; less important for Teamcenter production use.

Sequential write performance:

- Oracle LGWR writes, temporary segment writes, tablespace creations; important to Teamcenter if archive logging is enabled.

RAID Configurations

Oracle has specific recommendations for RAID. If you have an Oracle MetaLink account, there are numerous articles concerning RAID performance. Find the following doc IDs under advanced search:

Doc ID 38281.1 - RAID and Oracle - 20 Common Questions and Answers
Doc ID 30286.1 - I/O Tuning with Different RAID Configurations

at the following URL:

metalink.oracle.com

Table 5-5, RAID Configuration Rankings[55]

Oracle File	Teamcenter Use	No RAID	RAID 0	RAID 1	RAID 0+1	RAID 3	RAID 5
Control File	✓	**2**	**1**	**2**	**1**	**5**	**3**
Redo Log File	✓	**4**	**1**	**5**	**1**	**2**	**3**
System Tablespace	✓	**2**	**1**	**2**	**1**	**5**	**3**
DBWn intensive data files	✓	**1**	**1**	**2**	**1**	**5**	**5**
Indexed read-only data files		2	**1**	**2**	**1**	**2**	**3**
Sequential read-only data files		4	**1**	**5**	**1**	**2**	**3**
Direct load-intensive data files		4	**5**	**1**	**1**	**2**	**2**

Table 5-6, RAID Level Properties[56]

Raid Level Properties	Teamcenter Use	No RAID	RAID 0	RAID 1	RAID 0+1	RAID 3	RAID 5
Data protection	safety	**2**	**1**	**5**	**5**	**4**	**4**
Acquisition and operating costs	$ cost	**1**	**1**	**5**	**5**	**3**	**3**

5.2.7 *Other Oracle Performance and Scalability Considerations*

This section describes additional performance and scalability considerations for Oracle servers.

5.2.7.1 Data Migration (Import) Performance

Experience with a number of legacy data migration efforts by Siemens PLM Software Services has shown that adjusting Oracle initialization parameters can have a significant

[55] 1=Best Performance, 5=Worst Performance

[56] 1=Lowest, 5=Highest

impact on data import performance. If you have a significant amount of data to import, consider the following changes to your **init.ora** settings (Table 5-7).

Table 5-7, Oracle Initialization Parameter Settings for Mass Imports

Oracle Parameter	Bulk Migration Value	Daily Production Value
log_checkpoint_interval	0	10000
dml_locks	200	500
log_checkpoint_timeout	0	1800

Make these changes only while the imports are in process, then change back to the original production settings when imports are not being performed. Often, customers import new bulk data off hours when users are not typically logged in (for example, second or third shift). Restarting the instance with the import **init*sid*.ora** file after users have left for the day, and then again with the production **init*sid*.ora** file before they login the next morning, provides improved performance for both purposes. Or you can set these parameters interactively:

```
alter system set parameter parameter=value;
```

The settings in Table 5-7 are suggested starting points that may or may not provide benefit for your particular implementation. Make adjustments incrementally and carefully monitor performance and system utilization with each change to tune configuration for optimal performance. Taking these steps resulted in improving import throughput by as much as four times in some cases, after several iterations of careful, incremental changes and measurement.

5.2.7.2 Data Migration Export Considerations

If you typically export the database content with users actively using Teamcenter (or any other database activity) consider using the FLASHBACK_TIME clause in the export command. This parameter ensures that data is fully consistent for a specific export time. Without using the FLASHBACK_TIME clause customers with large database that take several minutes to export have reported anomalous behaviors after import importing the file into another database instance. See SFB-Teamcenter-6388 for details.

5.2.7.3 TCP/IP Registry Settings

For Oracle 10g r2 databases running on Windows, 3113 errors can be avoided by updating the registry of the Oracle clients (thin/rich client servers and command line utilities) to include the TcpMaxDataRetransmissions DWORD value with a value of 10. Note the default value is 5.

An excellent document with guidelines for Network Performance Tuning is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

See also section 10.2.2.4, *Network Resources*, for further information about improving network performance.

5.2.7.4 Suse

If you are not using Oracle's Automatic Memory Management (AMM), consider configuring HugePages on Suse systems. For large SGA sizes, HugePages can give substantial benefits in virtual memory management. For more information see Oracle Doc ID 361468.1.

Correspondingly, field experience has shown that disabling Transparent HugePages can also improve performance. See section 10.2.3.1 for information about disabling Transparent HugePages, also Oracle Doc ID 1557478.1.

5.2.8 Sizing Oracle Server for Teamcenter

Sizing guidelines presented in this section assume the performance tuning and configuration guidelines presented earlier in this chapter have been followed. How well the Oracle instance is tuned, as well as how memory, disk, and operating system components are configured has a direct impact on the system resource requirements of the Oracle server for Teamcenter.

NOTE

Both the environments and usage profiles used to simulate Teamcenter operations, including scenarios, transactions, number and types of files, operating systems and databases have changed since the previous Teamcenter Deployment Guide was published. ***For these reasons it is not valid to compare estimates in this revision of the guide with previous revisions.*** Use the simulation information in this chapter and extrapolate from that to estimate the target pilot environment.

5.2.8.1 Oracle Sizing Quick Reference

This quick reference is to help locate Oracle Server sizing information quickly for those who have already read this chapter. It assumes the reader has a good understanding of the factors that affect sizing information for the Oracle server and is familiar with terms defined later in the chapter (e.g. Oracle Demand Rate (ODR)) and in earlier chapters (Server Demand Rate (SDR), Usage Profiles, user categories, etc.).

The tables below provide guidelines for initial sizing estimates only. Do not use this information until you have reviewed this section in its entirety.

NOTE

Starting with TC 11.2.0, usage profiles for Thin and Rich Client are significantly different. The resource requirements for Thin and Rich Client are therefore not compatible at any tier.

Table 5-8, Oracle Server Sizing Information

Component	Guideline
CPU	Assuming the Usage Profile matches APA Benchmark;

Peak and average ODR per user for supported platforms is listed below as SPECint_rate2006 values:

	Thin Client		Rich Client	
Platform	**Peak SiR06 / User**	**Avg SiR06 / User**	**Peak SiR06 / User**	**Avg SiR06 / User**
AIX	0.009	0.004	0. 010	0.003
Solaris	0.006	0.002	0.004	0.001
Suse	0.019	0.009	0.011	0.006
Windows	0.028	0.014	0.012	0.005

Therefore, multiply the # of users by the average SDR and factor in an Operating Range Reserve of 20% to handle the login rate.

For example, using 3000 users on Suse with Oracle:

Step 1) **0.006** x 3000 = 18

Step 2) 18 ÷ 80% = 22.5

Step 3) Select a system with a SPECint_rate2006 rating of **22.5** or more.

Table 5-8, Oracle Server Sizing Information

Component	Guideline
Memory	Each *Concurrent* user consumes approximately the amount of RAM and SWAP listed below in Megabytes.

	Thin Client		Rich Client	
Platform	**MB RAM / User**	**MB SWAP / User**	**MB RAM/ User**	**MB SWAP / User**
AIX	17.1[57]	0.2	9.7[58]	0.4
Solaris	13.7	18.7	7.2	11.0
Suse	17.1	-	16.0	-
Windows	3.4	-	3.1	-

Note that Solaris pre-allocates SWAP space when a process is instantiated, even though no actual paging may occur.

5.2.8.2 Oracle Sizing Overview

Unless no other option is available, always install Oracle on a separate machine. It is much easier to optimize performance and scalability if separate application and database servers are implemented. For example, if Oracle and the Enterprise tier were installed on the same machine, changing a kernel parameter to improve the Enterprise server performance may adversely affect Oracle performance.

NOTE

It is strongly recommended that Teamcenter is the only schema installed in a single instance, i.e. the instance does not support other applications in addition to Teamcenter.

On UNIX systems an Oracle process is started for each *Connected* user. On Windows systems each user is handled by a thread within the Oracle process (oracle.exe).

Connected users typically consume less resource than *Active* users. Therefore logging in numerous users (connecting them) and taking resource measurements is not as accurate as measuring resource consumption during active use. The findings below are based on *Concurrent*, *Active* users.

[57] This is the actual RAM used with AME factor set to 2.0 (see section 4.5.3.2). Memory measured by SAR and other utilities will be twice this value.

[58] This is the actual RAM used with AME factor set to 2.0 (see section 4.5.3.2). Memory measured by SAR and other utilities will be twice this value.

5.2.8.3 Usage Profile

Customers routinely request sizing recommendations based purely on an estimate of the number of users they expect to access the system. Unfortunately, it is simply not possible to make an accurate recommendation based on user numbers alone. ***Which product features are used, and how often, is the predominant system sizing consideration.***

NOTE

You should obtain empirical system utilization data specific to your anticipated usage from your pilot deployment, and then extrapolate that to estimate the system resources needed for the production environment. A pilot implementation is highly recommended to obtain this empirical utilization data before sizing for production.

Refer to section 4.5.2, *APA Benchmark Usage Profile* for a full explanation of the APA Standard Rich Client usage profile used to estimate Oracle server sizing.

5.2.8.4 Oracle Server Sizing

Oracle server sizing for Teamcenter varies depending upon the anticipated *Usage Profile* (refer section 4.5.2), client type, and server hardware platform.

Table 5-9, Rich Client Oracle Server CPU, Memory, and Swap Usage illustrates per user resource consumption of the Oracle server hosted on various hardware / OS combinations, as measured in the Teamcenter APA Performance and Scalability Lab with the standard APA Rich Client usage profile. For comparative purposes, CPU utilization is represented in SpecINT_Rate2006 (SiR) values and rounded up to one thousandth, per user.

Overall, average CPU and RAM requirements for the Oracle Server with Teamcenter 11.2.1 have increased noticeably from the baseline release, except for AIX CPU, which mysteriously dropped. AIX, Suse, and Windows show very little to no swap / pagefile use.

Table 5-9, Rich Client Oracle Server CPU, Memory, and Swap Usage

Teamcenter Version	Platform	Number of Users	Peak SiR06 / User	Avg SiR06 / User	MB RAM / User	MB SWAP / User
Tc 10.1	AIX	500	0.027	0.010	6.3	0.4
	Solaris	2000	0.003	0.001	6.5	10.0
	Suse	1500	0.011	0.005	15.4	-
	Windows	2000	0.008	0.003	3.3	-
Tc 11.2.1	AIX	500	0.010	0.003	9.7	0.4
	Solaris	2000	0.004	0.001	7.2	11.0
	Suse	1500	0.011	0.006	16.0	-
	Windows	3000	0.012	0.005	3.1	-
Deltas	AIX		-61.5%	-68.6%	54.5%	-
	Solaris		50.0%	19.4%	9.7%	10.1%
	Suse		-	14.5%	4.3%	-
	Windows	1000	58.0%	90.5%	-6.8%	-

5.2.8.5 Oracle Memory Sizing for Teamcenter

Figure 5-3 illustrates memory consumption of the Oracle server on a single IBM System x3250 M3 running Linux with 32 GB ram for the APA standard Rich Client usage profile. This profile results in an Oracle client process[59] for each of the 1500 logged in users. Each running tcserver process on the Enterprise tier results in a corresponding Oracle client process running on the Oracle server. Dividing the total memory used by number of Oracle clients yields memory required per process. With this profile, each Oracle client consumes approximately 16.0 MB of physical ram on Linux, or 16 MB per *concurrent* user (your *Usage Profile* may be significantly different).

[59] On Windows, Oracle client are implemented as threads.

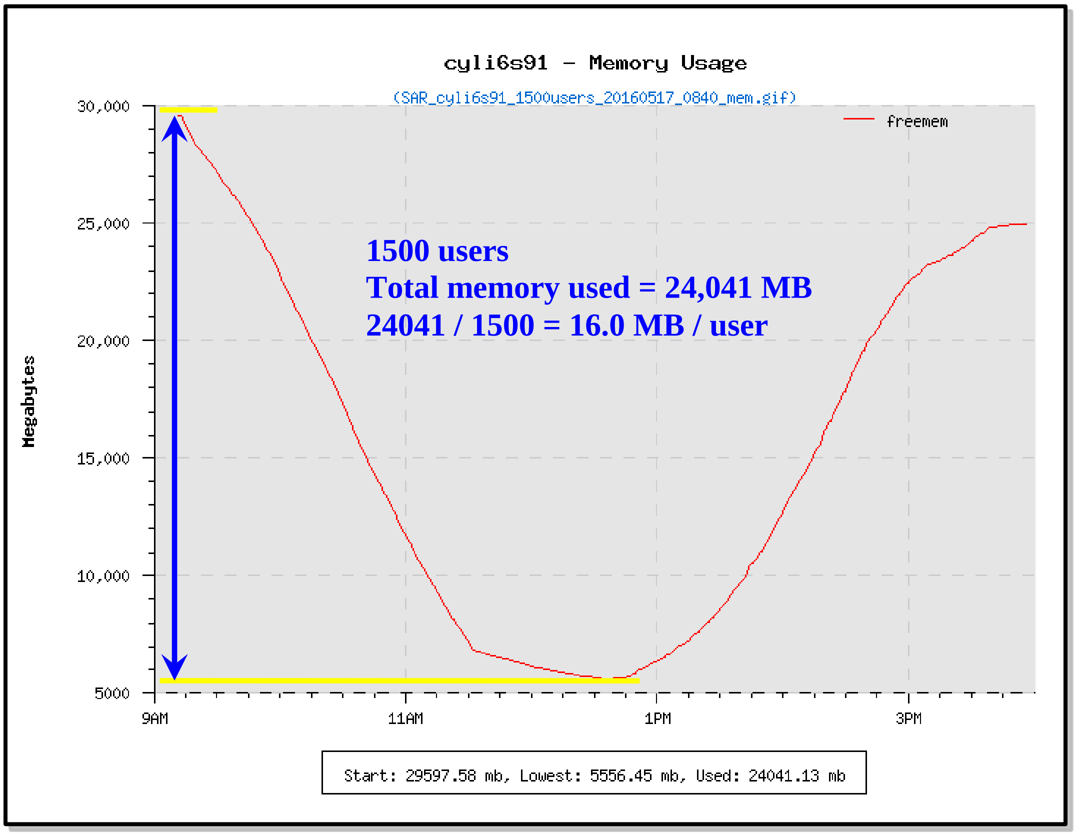
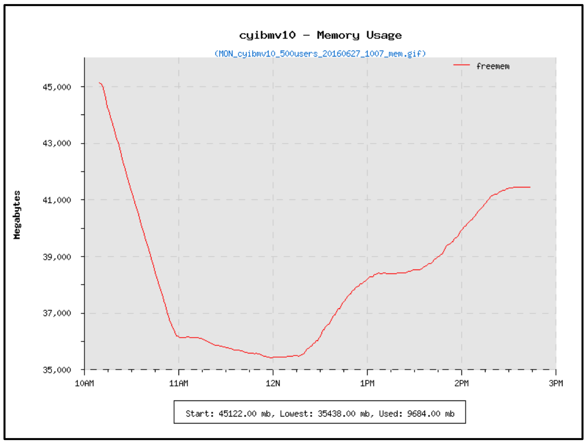


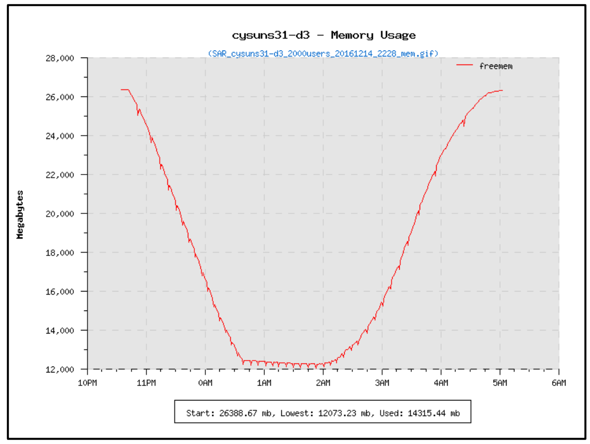
Figure 5-3, Oracle Server RAM Utilization – Linux

As shown in Figure 5-4, with the APA standard Rich Client usage profile, Oracle memory consumption is very similar across all platforms except Windows. As more users log in, Oracle consumes more RAM. On Windows, Oracle uses threads instead of child processes to support each user resulting in a lower memory requirement than UNIX deployments. These charts track available memory as users login, work, and then logout (freemem on UNIX). As more users log in, more RAM is consumed. Used memory is equal to the starting amount before the benchmark starts, less the lowest amount observed (Used = Start - Lowest).

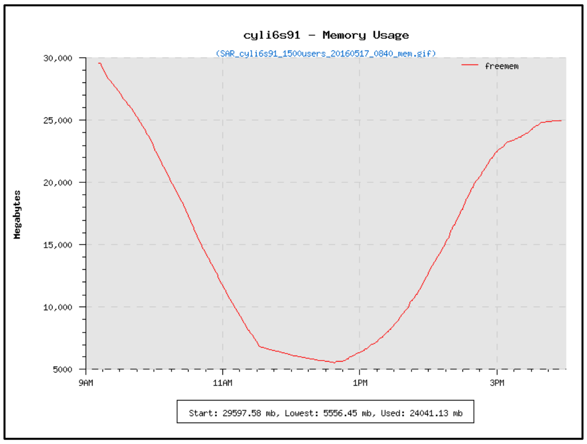
AIX – 500 users[60]



Solaris – 2000 users



Suse – 1500 users



Windows – 3000 users

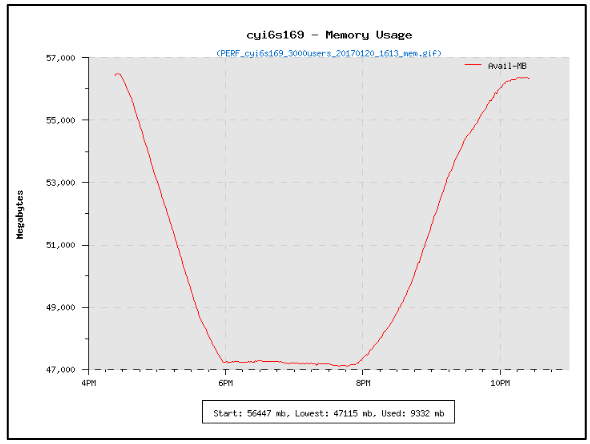


Figure 5-4, Rich Client Oracle Server RAM Utilization – All Platforms

[60] This is the value of RAM measured with SAR. The actual RAM used with AME factor set to 2.0 will be half this value (see section 4.5.3.2).

As seen in Figure 5-5 for this release of Teamcenter, Oracle memory has increased somewhat from the baseline 10.1 release except for AIX, which increased significantly, and Windows, which decreased somewhat. The red columns represent RAM, the blue columns SWAP/page. Note that AIX, Suse, and Window use little to no SWAP / pagefile space if there is ample physical memory.

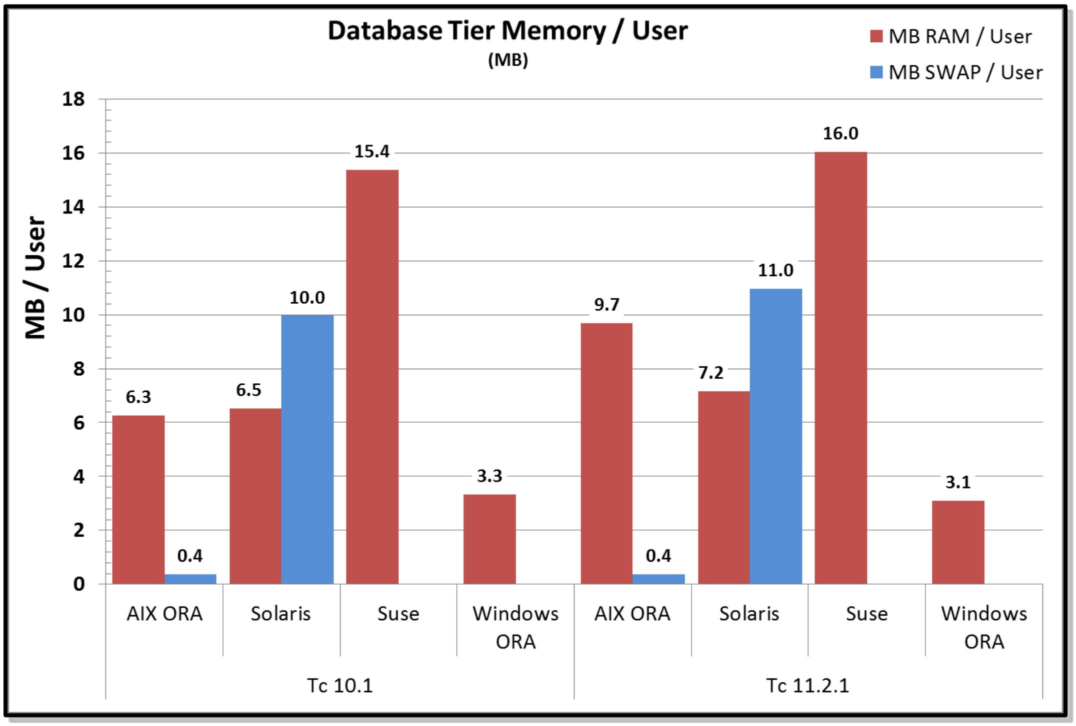


Figure 5-5, Rich Client Oracle Server per User Platform Memory Utilization

Based on these measurements, Table 5-10 provides an estimate of how many users an Oracle server can support for the given amount of physical memory installed, with usage profiles similar to the APA standard Rich Client usage profile. These estimates assume adequate CPU capacity to support these numbers of users. The *Estimated # of Users* columns represent an approximate number of users the system can support before paging would occur. Above that number of users the system may continue to function but with increased end-user response times due to paging.

The projections in Table 5-10 are based on APA standard Rich Client usage profile and assume each OS requires approximately 2 GB ram. These estimates do not include SGA requirements (see section 5.2.3.5 for SGA / PGA guidelines).

Table 5-10, Oracle Server Memory Estimates

		AIX	Solaris	Suse	Windows
RAM per User		**9.7 MB**	**7.2 MB**	**16.0 MB**	**3.1 MB**
Physical Memory (GB)	**Usable Memory (GB)**	**Estimated Number of Users**			
4	2	210	280	120	650

8	6	630	850	380	1970
12	10	1050	1430	630	3290
16	14	1480	2000	890	4600
24	22	2320	3140	1400	7240
36	34	3170	4860	2170	11190
48	46	4860	6580	2930	15140
64	62	6550	8860	3960	20400
96	94	9930	13440	6000	30940

5.2.8.6 Oracle SWAP Sizing for Teamcenter

NOTE

Nearly all contemporary operating systems use virtual memory paging to accommodate programs that require more memory than is physically available, although some still do swap out entire programs under extreme memory pressure. However the term '*swap*' is typically still used with UNIX based operating systems even though memory is being paged.

As with the Enterprise tier, only Solaris requires swap space to be allocated as Figure 5-6 shows.

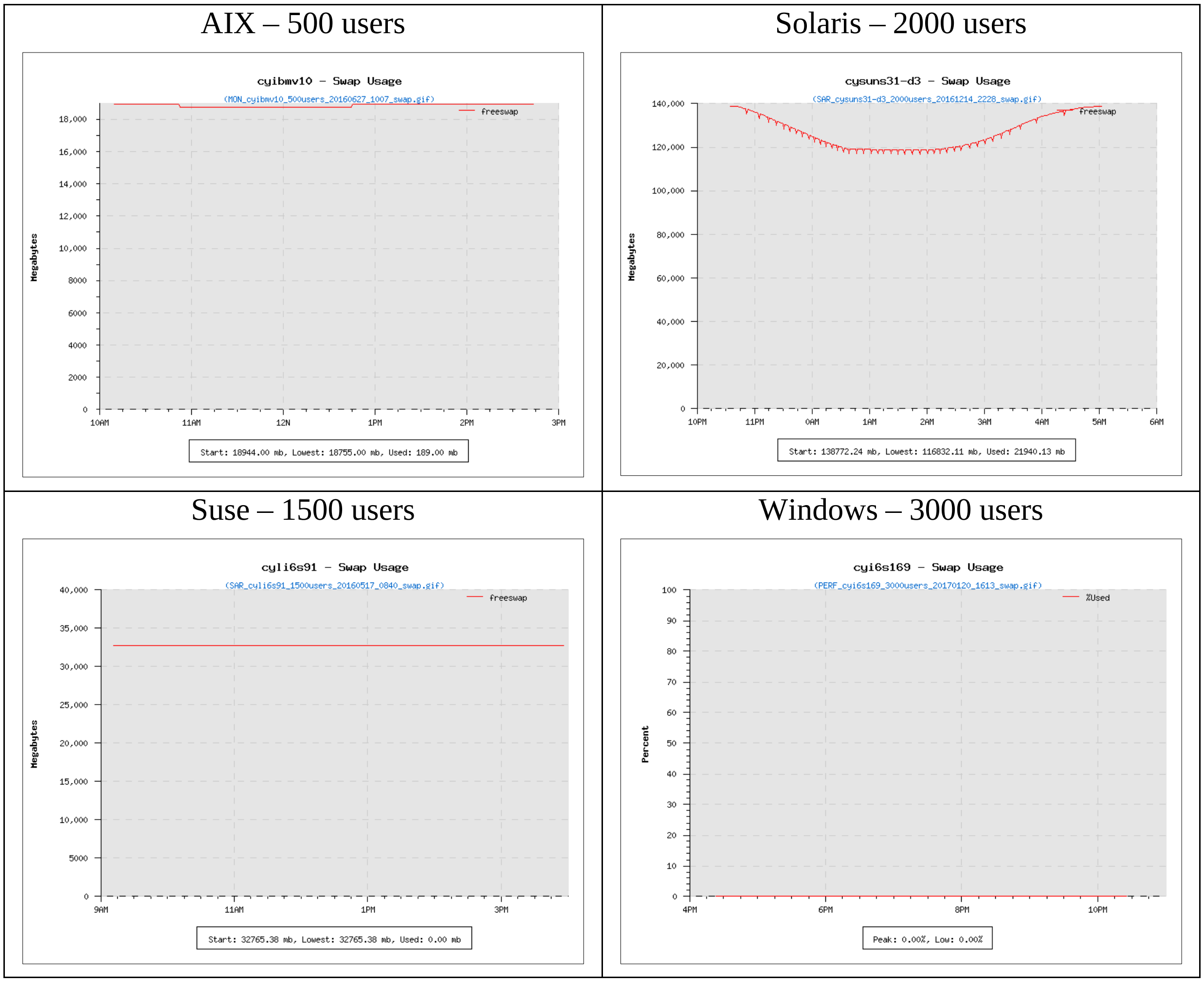


Figure 5-6, Rich Client Oracle Server SWAP / Page Utilization – All Platforms

5.2.8.7 Oracle CPU Sizing for Teamcenter

If the Demand Rate can be determined (i.e. how many SPECint_rate2006s are needed) for the expected Teamcenter Oracle *Usage Profile* it can be cross-referenced against various systems listed with SPEC to find one that is suitable. The value derived for the Oracle usage profile is referred to as the Teamcenter Oracle Demand Rate (ODR).

See also section 4.5.3.4 above for additional information about determining server demand rates.

Determining the Teamcenter Oracle Demand Rate (ODR)

The *Usage Profile* plays a big role in determining the ODR and ultimately the size of the Oracle server. Oracle server sizing factors are again related to the types and frequency of Teamcenter operations but are generally more affected by:

- The amount of data managed
- The number of concurrently logged in users

- The data access patterns of users
- Settings in the web server configuration
- Database index and optimizer maintenance

If users are busy performing queries more CPU will be consumed so it is important to determine typical user activities to estimate how much CPU will be needed.

Before sizing a system, consider the target operating range acceptable at peak load. Measurements show that if the Oracle server is nearly 100% utilized there will be negative impact on user response time. See also section 4.5.4, *Impact of Overloading Enterprise Server CPU* for additional information. A more conservative maximum range of 80% should be considered.

Figure 5-7 shows CPU utilization for 1500 users using the APA standard Rich Client usage profile. The SPECint_rate2006 rating of the database server used for this benchmark is 106.9. At 7.9% CPU average and 15% peak, this equates to an Oracle Demand Rate (ODR) per user of 0.006 (107 x 0.079 ÷ 1500) at steady state (average) and 0.011 at peak (107 x 0.15 ÷ 1500).

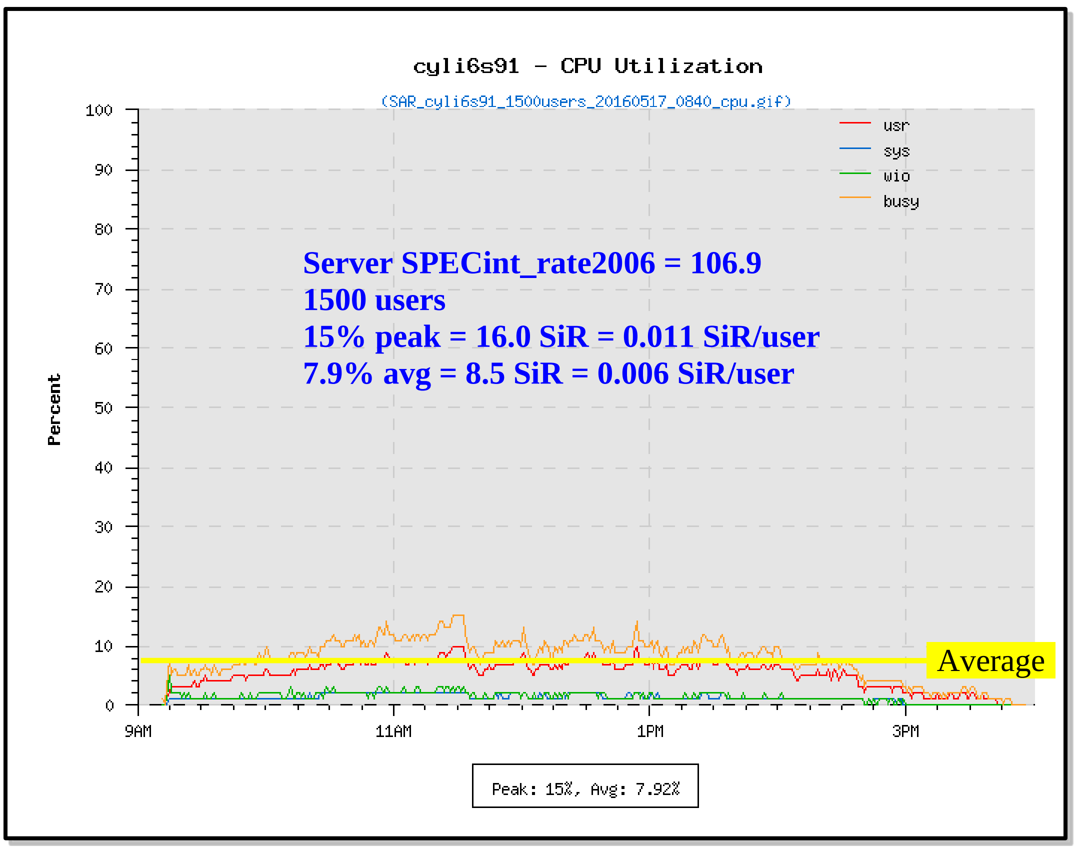
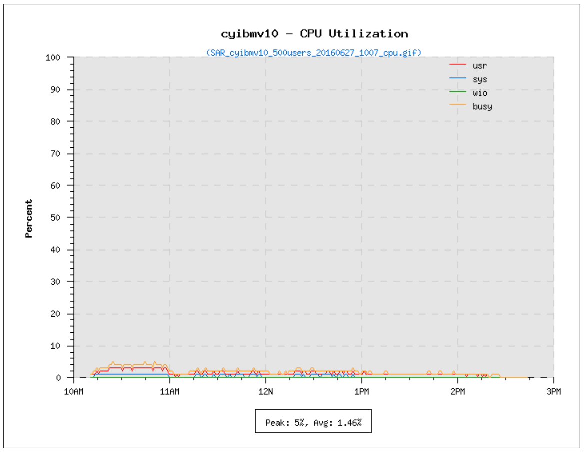
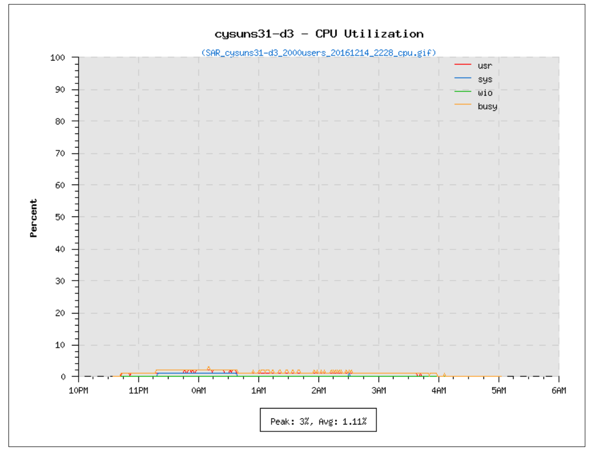


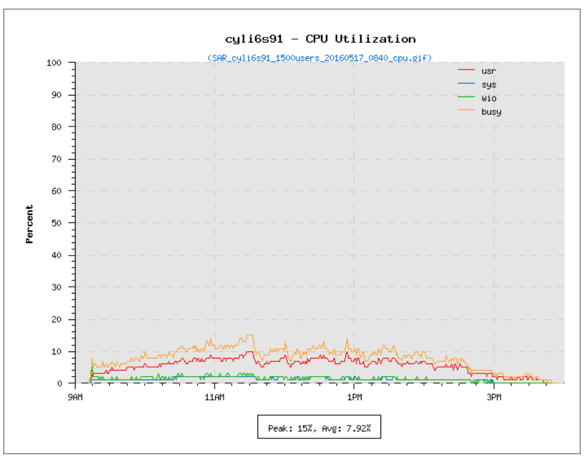
Figure 5-7, Rich Client Oracle CPU Utilization – Linux

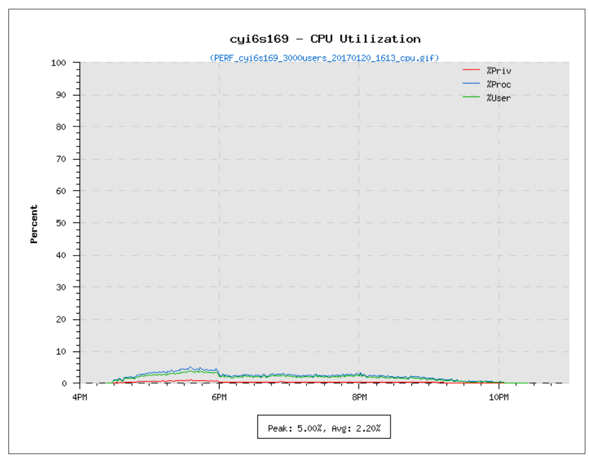
In order to keep CPU utilization below 80% for this Rich Client usage profile, a system with a SPECint_rate2006 rating of 20 or better is needed to serve Oracle for this profile (16 / 80%).

Figure 5-8 shows Oracle CPU utilization for all measured platforms.

<table>
<tr><td>AIX – 500 users</td><td>Solaris – 2000 users</td></tr>
<tr><td>

(2 x POWER 7+ @ 4.116 GHz)</td><td>

(6 x SPARC-T7 @ 4.133 GHz)</td></tr>
<tr><td>Suse – 1500 users</td><td>Windows Oracle– 3000 users</td></tr>
<tr><td>

(4 x Intel® Xeon® X3440 @ 1.2GHz)</td><td>

(16 x Intel® Xeon® E5-2640 v3 @ 2.6GHz)</td></tr>
</table>

Figure 5-8, Rich Client Oracle Server CPU Utilization – All Platforms

All platforms show notably more CPU required than the baseline release as seen in Figure 5-9. The blue portion of the columns indicates average CPU, the red portion peak CPU.

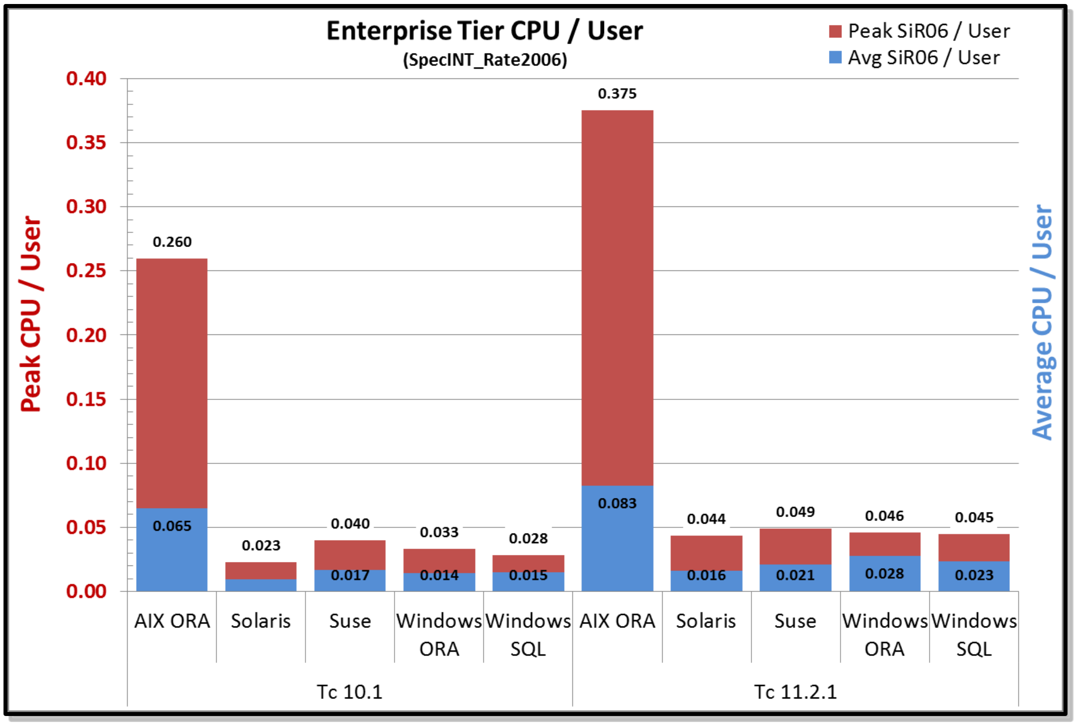


Figure 5-9, Rich Client Oracle Server per User CPU Utilization vs. Baseline

5.2.8.8 Oracle Disk Space Sizing

Three categories that consume disk space need to be considered:

- Swap requirements (see section 5.2.8.6 above)
- Oracle Installation - disk space requirements
- Teamcenter - database tablespace requirements

Oracle Installation

Refer to the Teamcenter *Installation on UNIX and Linux Servers* or *Installation on Windows Servers* manuals for space requirements of Oracle binaries.

Database Tablespaces

A key consideration for new Teamcenter deployments is how large to make database tablespaces. This is of course entirely dependent on how much information will be managed and will vary significantly for each implementation.

Very large implementations can require tens or even hundreds of gigabytes for Teamcenter data tablespaces, but typical implementations are on the order of a few gigabytes.

The database used for the APA Standard Rich Client usage profile benchmark (APA$_{DB}$_5K) consisted of 5,000 users with >800,000 items/item revisions, >1,000,000 datasets, 2.8M files, as well as TcSE data. An additional 45,000 users without data (except for home and inbox folders) are defined in this database. It also includes product structures with >130,000 bom lines, and over one million design elements objects. This database consumes nearly 60 GB of disk space in addition to system, temporary, and undo tablespaces as seen in Table 5-11.

Table 5-11, Oracle Server Tablespace Requirement for 5,000 Users with Data[61]

Tablespace Allocations and Space Used

Select	Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)
◉	IDATA	59,840.0	58,446.2	97.7
○	ILOG	544.0	381.5	70.1
○	INDX	128.0	1.0	0.8
○	PERFSTAT	256.0	1.0	0.4
○	SYSAUX	800.0	624.4	78.1
○	SYSTEM	512.0	399.7	78.1
○	TEMP	3,584.0	2.0	0.1
○	UNDOTBS1	3,488.0	17.3	0.5

Sample of Large and Small tables

Table	Rows
pom_backpointer	48,198,598
ppom_object	26,609,648
pworkspaceobject	7,420,034
pform	3,045,090
pimanrelation	2,949,760
pimanfile	2,831,074
pnote_types	1,787,717
pdataset	1,060,238
pcpd0designiteminstance	1,022,472
pcpd0designmodelelement	1,022,472
pcpd0designmodelelement	1,022,472
ppsoccurrence	925,042
ppsoccurrencethread	925,042
pitemrevision	823,895
pitem	823,893
previsionanchor	529,661
pugpartboundingbo	344,374
ppsbomview	133,174
pperson	50,014
puser	50,014
pappearance	47,872
pimantype	8,212
pepmjob	40
pimanvolume	20

If large amounts of data will be imported with the initial deployment, consider increasing these settings using the Oracle Enterprise Manager. As always, a pilot implementation will help determine actual database tablespace requirements.

ilog Tablespace

The Teamcenter Audit Manager can potentially require a large amount of disk space. This feature allows user execution of various actions to be recorded (logged) in Oracle tables. Without diligent management, these tables can grow to hundreds of gigabytes over time. Also, recording Audit Manager event records to the database can increase CPU on the Oracle server 10% or more. Recording Audit Manager event records to a file has little impact on Oracle server CPU.

Review carefully the business requirements to log Teamcenter actions and keep Audit Manager disabled if possible (the TC_audit_manager preference default). If auditing is a

[61] This is a 50,000 user database, the first 5,000 of which have roughly 100 items with datasets. The remaining 45,000 users have only home and inbox folders.

business requirement, establish a practice of writing Audit Manager data to files with the audit_archive utility, and archiving those files to backup media.

5.2.9 *Other Considerations*

5.2.9.1 BIOS settings

Processor Power States

Some processors support multiple power states intended to conserve energy when that is important. If performance is critical to your deployment, consider disabling C-States (processor states) and / or P-States (performance states) in the BIOS settings. Not all processors support these options, and not all BIOS implementations support configuring them. Refer to the hardware user’s guide for your particular servers.

5.2.9.2 Improving Teamcenter Database Write Performance

Setting the TC_POM_CNW environment variable in the pool server tc_profilevars file can improve database performance for write operations (e.g. create, update, delete…) by adjusting Oracle’s COMMIT_WAIT and COMMIT_LOGGING functionality. After setting the environment variable new Teamcenter sessions will have COMMIT_WAIT=NOWAIT and COMMIT_LOGGING=BATCH set in their environment.

This functionality can help with applications that perform a huge amount of insert or update activity; this functionality will not help query operations. The CATIA integration SWIM interface, and PLMXML imports are good candidates for this functionality.

Setting TC_POM_CNW to *any value* enables the associated functionality, e.g.:

```
Se TC_POM_CNW=TRUE
```

The environment variable must be removed from the execution environment to disable this functionality.

NOTES:

1. This functionality must only be set by an expert that understands all of the database implications.
2. This functionality has shown to have little benefit when used with high speed disks.
3. Refer to Oracle’s documentation on COMMIT_WAIT and COMMIT_LOGGING for additional information.
   http://docs.oracle.com/cd/E24693_01/server.11203/e24448/initparams032.htm
   http://docs.oracle.com/cd/E24693_01/server.11203/e24448/initparams030.htm

5.2.10 *Configuring Teamcenter for Oracle RAC*

5.2.10.1 Oracle RAC Overview

Oracle Real Application Clusters (RAC) provide a highly fault-tolerant database resource tier with excellent load balancing on a variety of OS platforms for Teamcenter deployments with high availability (HA) requirements. Scalability benchmarks to 5,000 concurrent users conducted on hardware clusters with up to two nodes have shown reliable operation and fast failover with overall server response times somewhat slower than a single node, non-RAC database instance. When considering deploying Teamcenter with an Oracle RAC database end user response times should be evaluated with the target Oracle RAC systems to ensure performance is consistent with the users' expectations.

There are essentially no installation or configuration changes required to Teamcenter other than ensuring the appropriate Oracle configuration files are created for the cluster instance using the Oracle DBCA. When Oracle RAC is properly configured and operating, Teamcenter sees the database as a single instance during installation, setup, upgrade, and normal operation. Using Oracle RAC as the database base tier for Teamcenter requires only:

- A properly configured and functioning hardware cluster, such as Sun Solaris Cluster 3.2 software (Optional)
- Properly installed and configured Oracle Clusterware software
- Properly installed and configured Oracle RAC software
- A properly configured and functioning Oracle RAC database

This section assumes the hardware cluster, Oracle Clusterware, and Oracle RAC software are properly installed and running. Consult with your platform provider or Oracle for support installing and configuring those components; many have published guides to install / configure their cluster hardware / software and configure for Oracle RAC[62]. Once those components are properly installed and functioning properly, configuring Teamcenter for Oracle RAC consists of the following activities:

- Configuring the Oracle RAC Database with DBCA
- Configuring additional Oracle remote and local listeners in the tnsnames.ora file
- Configuring the Oracle Net Services Database Connection in the tnsnames.ora file

5.2.10.2 Configuring Oracle RAC Database

Oracle RAC installation and configuration procedures for Oracle 11+ have changed significantly since the previous revision of this guide. Consult the latest Oracle

[62] e.g., *Installation Guide for Solaris Cluster 3.2 Software and Oracle® 10g Release 2 Real Application Clusters* *Fernando Castano, June 2007, © Sun Microsystems, Inc.*, provides detailed step-by-step instructions for the Solaris platform.

documentation to install and configure Oracle RAC. Additional guidelines will be provided here as they are developed.

5.2.10.3 *Oracle RAC Sizing and Tuning Considerations*

Oracle RAC has specific sizing and tuning considerations.

CPU Capacity

Oracle RAC systems imposes some overhead on Teamcenter when using multi-node clusters. As system nodes are added to an Oracle RAC cluster, the overhead required to keep all SGA caches synchronized increases. This overhead varies depending on the platform on which it is deployed. For example on Oracle SuperCluster CPU overhead increased linearly as nodes were added and did not exceed 26% peak or 38% average with four nodes. See Figure 5-10. Other platforms have not been measured and CPU requirements could vary significantly.

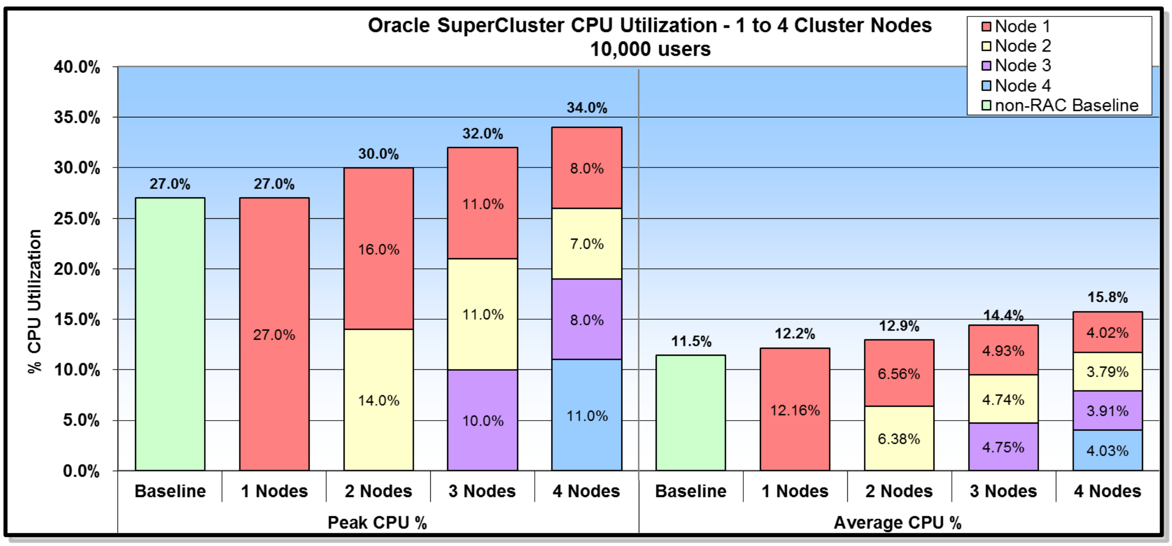


Figure 5-10, Example Oracle SuperCluster CPU Utilization

Cluster Interconnect

Benchmarks showed that the cluster-interconnect *latency*, and not the bandwidth, has the most impact on server response times as the number of cluster nodes is increased. Be certain to keep cluster interconnect latency as low as possible; sub-millisecond if possible. An InfiniBand cluster interconnect provides very low latencies between nodes for optimal Cache Fusion processing and Global Cache Service (GCS) operations.

Permanently Cached Tables

Evidence suggests that permanently caching frequently used tables (i.e. 'pinning tables') can have a detrimental impact on Oracle RAC performance. If you had previously pinned tables for a non-RAC instance un-cache the tables when migrating to Oracle RAC by setting the table(s) to use the 'default' buffer pool.

To remove tables from the keep pool:

```
alter table tablename storage(buffer_pool default);
```

Index Structure

Benchmarks showed that certain indexes benefit from alternative structures (vs. the standard b-tree). If migrating an existing Teamcenter installation reference Software Field Bulletin 10228:

5.2.10.4 Other Oracle RAC Considerations

If you are running Oracle in archive mode, note that archive logs need to be readable by all nodes in the cluster. This is because the recovering node (which may be any) will need to access the archive logs to perform the roll forward for recovery. If you configure these logs go to a shared disk just make sure all nodes are attached. If you configure them for local disk ensure the directory is exported with NFS mounts to the other nodes.

5.3 Managing Microsoft SQL Server Database Servers

NOTE

In collaboration with Siemens PLM Software, Microsoft has published a *Best Practices for Running Siemens Teamcenter on SQL Server* guide available *here*.

This section provides the necessary information to initially size and configure an SQL Server database instance for Teamcenter. Additional troubleshooting information for SQL Server performance issues is available from Microsoft at *http://technet.microsoft.com/en-us/library/dd672789(v=sql.100)*. Although written against SQL Server 2008, most should be applicable to later versions as well.

5.3.1 *SQL Server Performance Tuning Quick Reference*

This section is to help quickly validate Microsoft SQL Server configuration settings for Teamcenter. As with the preceding section, it assumes the reader has a reasonable understanding of the factors that affect performance and is familiar with terms defined throughout this chapter. Tuning any database management system is an iterative process requiring careful record keeping and patience to measure, make configuration changes, and measure again, until optimal performance is achieved.

The tables below provide guidelines for initial configuration and performance tuning only when used with Teamcenter. Do not use this information until this section has been reviewed in its entirety.

Table 5-12 lists a number of SQL Server configuration settings and tuning methods that have the greatest impact to Teamcenter system performance. These should be checked to be certain they are correctly set or performed.

Table 5-12, SQL Server Performance Tuning and Configuration Checklist

Description	Location / Notes
For large user deployments (>1000) with SQL Server 2008 R2 be sure to install Cumulative Update 7 (CU7). SQL Server 2012 and later versions includes this update.	
Windows operating system parameters • For 32-bit systems with more than 4GB RAM, verify /pae is set • Verify that /3GB is NOT set if memory exceeds 16GB	Section 5.3.4.4

Description	Location / Notes
Enable AWE support for 32-bit CPUs within SQL Server • sp_configure 'awe enabled', 1 RECONFIGURE GO	Section 5.3.4.1 Reference: http://support.microsoft.com/default.aspx?kbid=274750
Tune and monitor SQL Server • SQL Server tuning parameters • Windows Server tuning parameters • Teamcenter indexes for SQL Server • dbcc checkdb	Section 5.3.4.1
Optimally configure database files on disks with special consideration for tempdb (SQL Server's work area) and the transaction logs	Section 5.3.4.2 Disk Resources (pg.132)
Disable slow SQL reporting except to diagnose performance issues: • TC_SLOW_SQL=-1	See Slow SQL Reporting

5.3.2 *SQL Server Performance*

This section provides information generally applicable to most customer MS SQL Server deployments. There are very few settings required in SQL Server to tune for performance. However there are many monitoring tools available to monitor MS SQL Server performance. For more information on the available monitors, see *HOW TO: Troubleshoot Application Performance Issues* on the Microsoft support site.

One of the most readily accessible tools to monitor system wide resource utilization by the SQL Server is Windows Performance Monitor (PerfMon). A system with SQL Server installed locally will have numerous SQL Server specific performance counters available to PerfMon. With PerfMon, performance data can be accessed real-time or saved to a log and analyzed at a later time. Figure 5-11 shows several of the SQL Server specific metrics that can be monitored[63].

[63] When properly configured, PerfMon can access system utilization data on remote machines as well as local data.

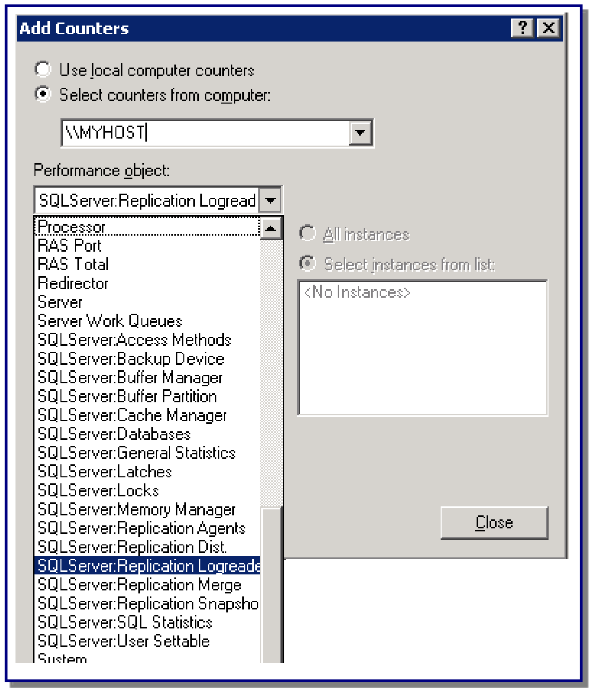


Figure 5-11, PerfMon SQL Server Metrics

Performance log files generated by PerfMon can later be read back into PerfMon for charting, or if the file is specified as .csv (comma separate values) format when created, it can be loaded into other applications, such as a spreadsheet.

5.3.2.1 Teamcenter SQL Server Performance Issues

Many SQL Server performance issues are related to improperly sized hardware. Refer to section 5.1.1, *Properly Sizing the Server* at the beginning of this chapter for information about properly sizing the database server.

Operating System Considerations

There are no known Operating System specific considerations for SQL Server 2008 or later. See however section 5.3.4.4 for network parameter guidelines.

5.3.3 *SQL Server Indexes*

The following indexes were found to reduce database deadlocks (which affect performance) and/or reduce CPU / Disk I/O.

Table 5-13, Teamcenter Indexes for SQL Server

Command
The following can be added to Tc 11.2.x deployment, and may not be included standard in later releases.
CREATE INDEX [new_index_53_52_PPOM_MEMBER] ON [M08TCwr_001].[dbo].[PPOM_MEMBER] ([rgroupu]) INCLUDE ([puid])

Table 5-13, Teamcenter Indexes for SQL Server (continued)

Command
CREATE INDEX [new_index_51_50_PGROUPMEMBER] ON [M08TCwr_001].[dbo].[PGROUPMEMBER] ([rroleu]) INCLUDE ([puid])
CREATE INDEX [new_index_6_5_PIMANTYPE] ON [M08TCwr_001].[dbo].[PIMANTYPE] ([pisAbstract]) INCLUDE ([puid])
The following may or may not have an impact depending on the specific usage profile. If syslogs shows a significant number of database deadlocks, or if server CPU or disk I/O appear to be issues, add the indexes one at a time and measure if there is an improvement (or regression).
install -add_index infodba <passwd> dba CUSTOM_TYPE_class 0 imantype type_class
install -add_index infodba <passwd> dba CUSTOM_PROPINFO_bobject 0 propertyinfo business_ object
install -add_index infodba <passwd> dba CUSTOM_USER_status_license 0 POM_user status license_level
install -add_index infodba <passwd> dba CUSTOM_INBOX_owner 0 TaskInbox owner

5.3.4 *SQL Server Configuration and Maintenance*

There are relatively few configuration and maintenance issues to address with SQL Server other than those presented in the *Database Configuration and Maintenance* section earlier in this chapter.

5.3.4.1 SQL Server Tuning

SQL Server 2008 R2 has automated many of the tuning requirements from previous releases. To enable these set the following parameters in the SQL Server Management Console.

Table 5-14, SQL Server 2008 R2 Tuning Parameters

Parameter	Setting (command)
Max Degree of Parallelism	1 (sp_configure 'max degree of parallelism', 1; reconfigure;)
named-pipe communication	Disable (this is the default value and should be left disabled)
AUTO_UPDATE_STATISTICS	ON (alter database @DB_NAME@ set AUTO_UPDATE_STATISTICS_ON)
AUTO_UPDATE_STATISTICS_ASYNC	ON (alter database @DB_NAME@ set AUTO_UPDATE_STATISTICS_ASYNC_ON)

If a very large buffer cache (> 4 GB) is required for 32 bit versions of SQL Server, enable AME as follows:

```
sp_configure 'awe enabled', 1
RECONFIGURE
GO
```

This is not required for 64 bit SQL Server. Note this configuration option is deprecated in SQL Server 2008 R2 and is removed from SQL Server 2012 and later. See http://support.microsoft.com/kb/2644592 .

To minimize contention for frequently updated objects, see SFB-Teamcenter-6327 to set READ_COMMITTED_SNAPSHOT to ON.

Occasionally check database integrity database by performing the following:

```
dbcc checkdb
```

See http://msdn.microsoft.com/en-us/library/ms176064.aspx for more information about checkdb.

Fragmentation occurs when density falls below 80% for large tables. Please see *Reorganizing and Rebuilding Indexes* on *technet.microsoft.com* for more on information on de-fragmenting and rebuilding indices.

5.3.4.2 SQL Server File System Optimization

Under some usage profiles performance can be improved and CPU utilization reduced somewhat on multi-processor systems by increasing the number of tempdb data files to equal the number of physical processor cores. All files should be equally sized, both initially and within their growth segments. These files may reside on the same disk. Measure the performance and CPU utilization before and after this change to ensure a positive effect.

It's worth noting as technology advances and allows for more and more cores on a single processing chip, the number of tempdb data files should not necessarily match the number of CPUs on a system. For systems running more than 8 cores, 8 tempdb files should be sufficient.

As always, measure the impact to performance of any database changes for your particular usage profile. If there is tempdb contention, consider adding more. You may want to experiment with the number of tempdb files defined to find the optimal configuration.

See also the section concerning *Disk Resources* earlier in this chapter.

5.3.4.3 SQL Server Table Organization

As with other database options, some additional performance improvement may be possible in SQL Server with careful management of table organization. Splitting data tables across multiple LUNS and/or controller may also improve performance for large databases. In SQL Server, one can use user-defined file-groups to specify which data tables and are in which data files.

One may also add additional data files to the teamcenter database. If you choose to do this, take caution when adding the files to ensure they are being placed on the appropriate drives. In some cases, the new files will reap a performance benefit by placing them on a new LUN. It is recommended to keep the data files separate from the log and tempdb files.

5.3.4.4 Windows Server Parameters for SQL Server

The following Windows Server 2008 R2 network parameters were found to reduce network bottlenecks that might limit scalability.

Table 5-15, Network Parameters for SQL Server

Parameter	Value	Location / Command / Purpose
TcpTimedWaitDelay	0x1E (30s)	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\(REG_DWORD) Default value is 0xF0, which sets the wait time to 240 seconds (4 minutes). This will effectively hold the connection hostage for 4 minutes increasing the number of connections in use.
maxuserport	0x0000ffff (65535)	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\(REG_DWORD) This is especially important on the web tier as each user requires a separate ephemeral IP port.
Receive-Side Scaling State	disabled	netsh int tcp set global rss=disabled
Chimney Offload State	disabled	netsh int tcp set global chimney=disabled
NetDMA State	disabled	netsh int tcp set global netdma=disabled
Direct Cache Access (DCA)	disabled	netsh int tcp set global dca=disabled
Receive Window Auto-Tuning Level	normal	netsh int tcp set global autotuninglevel=normal
Add-On Congestion Control Provider	ctpc	netsh int tcp set global congestionprovider=ctcp
ECN Capability	disabled	netsh int tcp set global ecncapability=disabled
RFC 1323 Timestamps	disabled	netsh int tcp set global timestamps=disabled

5.3.5 Finding SQL Server System Bottlenecks

Refer to section 5.1.5, Diagnosing Database Bottlenecks earlier in this chapter.

5.3.5.1 SQL Server Performance Statistics

SQL Server 2008 R2 (and SQL Server 2005) includes a number of built-in performance reports. SQL Server 2008 and 2008 R2 have built in reports accessible via SQL Server Management Studio. You may also create and add your own custom reports. For more information, see http://msdn.microsoft.com/en-us/library/bb153684(v=sql.105).aspx

5.3.6 Sizing SQL Server for Teamcenter

Sizing guidelines presented in this section assume the performance tuning and configuration guidelines presented earlier in this chapter have been followed. How well the SQL Server instance is tuned, as well as how memory, disk, and operating system components are configured has a direct impact on the system resource requirements of the SQL Server for Teamcenter.

NOTE

You should obtain empirical system utilization data specific to your anticipated usage from your pilot deployment, and then extrapolate that to estimate the system resources needed for the production environment. A pilot implementation is highly recommended to obtain this empirical utilization data before sizing for production.

5.3.6.1 SQL Server Sizing Overview

Unless no other option is available, always install SQL Server on a separate machine. It is much easier to optimize performance and scalability if separate application and database servers are implemented.

NOTE

It is strongly recommended that Teamcenter is the only schema installed in a single instance, i.e. the instance does not support other applications in addition to Teamcenter.

Connected users typically consume less resource than *Active* users. Therefore logging in numerous users (connecting them) and taking resource measurements is not as accurate as measuring resource consumption during active use. The findings below are based on *Concurrent, Active* users. Refer to the description of the Usage Profiles in section 4.5.2, which used to derive this sizing information.

The SQL Server sizing data below was obtained from a series of scalability benchmarks[64] using Dell R510 with 8 x Intel® Xeon® X5550 processors @ 2.67GHz and 32 GB RAM. SQL Server 2012 was installed locally.

[64] Scalability benchmarks employ virtual users to simulate multiple users executing one or more specific usage profiles.

5.3.6.2 SQL Server Sizing

NOTE

Starting with TC 11.2.0, usage profiles for Thin and Rich Client are significantly different. The resource requirements for Thin and Rich Client are therefore not compatible at any tier.

SQL Server sizing for Teamcenter varies depending upon the anticipated *Usage Profile* (refer section 4.5.2), client type, and server hardware platform.

Table 5-16, Rich Client SQL Server CPU, Memory, and Swap Usage, illustrates per user resource consumption of the SQL Server hosted on a Dell 16 core Intel® Xeon® CPU E5-2640 v3 @ 2.60GHz running Windows Server 2008R2 with the standard APA Rich Client usage profile. For comparative purposes, CPU utilization is represented in SpecINT_Rate2006 (SiR) values and rounded up to one thousandth, per user.

Overall, average CPU and RAM requirements for SQL Server with Teamcenter 11.2.1 have increased from the baseline release.

Table 5-16, Rich Client SQL Server CPU, Memory, and Swap Usage – 3,000 Users

Teamcenter Version	Peak SiR06 / User	Avg SiR06 / User	MB RAM / User	MB Page / User
Tc 10.1	0.017	0.008	9.8	-
Tc 11.2.1	0.029	0.012	14.5	-
Deltas	73.8%	48.5%	47.6%	

5.3.6.3 SQL Server Memory (RAM) Sizing for Teamcenter

Figure 5-12 shows the memory utilization on the SQL Server system was measured at 14.5 GB executing the 3000 user Rich Client usage profile. This equates to 14.5 MB of physical ram per user, including user thread space and cache growth.

This chart tracks available memory as users login, work, and then logout. As more users log in and more data is accessed, more RAM is consumed. Used memory is equal to the starting amount before the benchmark starts, less the lowest amount observed (Used = Start - Lowest).

Note that SQL Server will allocate to its buffer cache as much memory as is available on the system, unless you specifically set an upper limit with *Maximum server memory* under *Server Properties*.

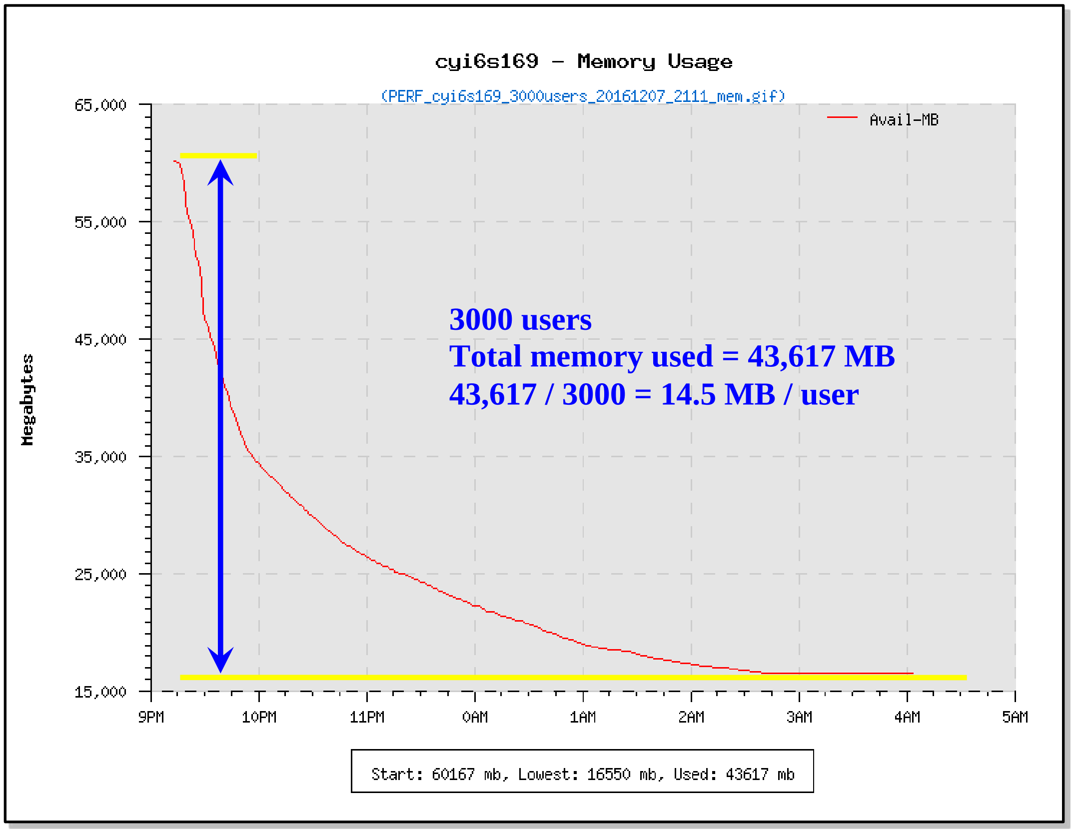


Figure 5-12, Rich Client SQL Server RAM Utilization Profile

For this release of Teamcenter, SQL Server memory usage has increased noticeably compared to the baseline release as seen in Figure 5-13. The red columns represent RAM; there is zero pagefile utilization.

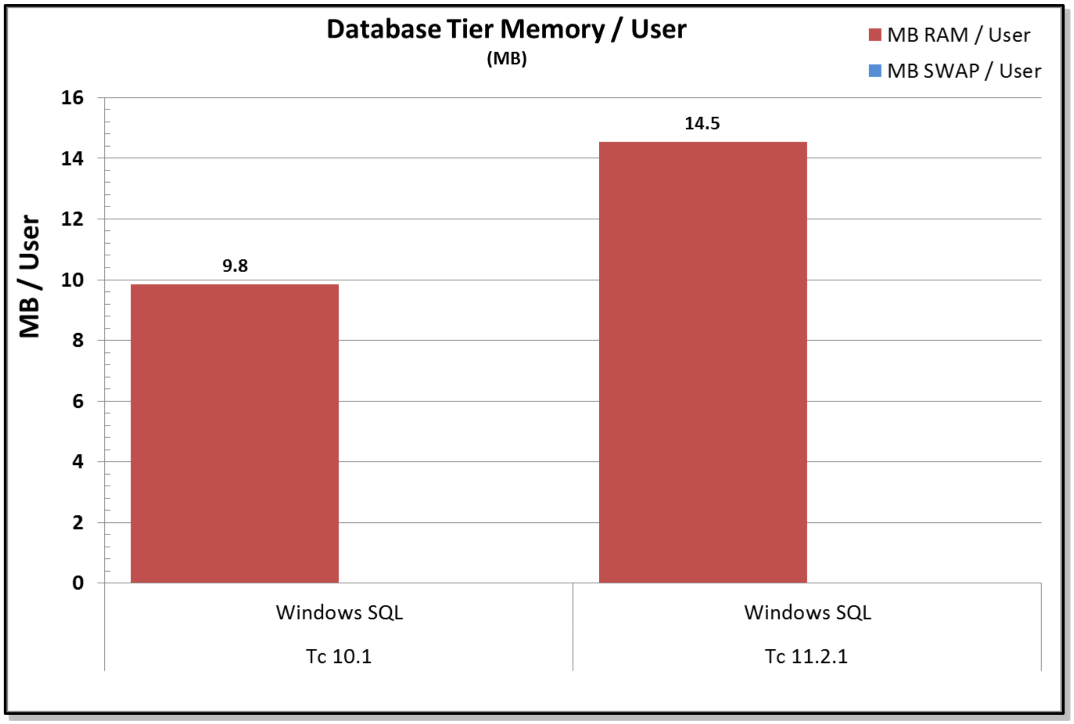


Figure 5-13, Rich Client SQL Server RAM Utilization vs. Baseline

The projections in Table 5-17 are based on APA standard Rich Client usage profile and assume each OS requires approximately 2 GB ram.

Table 5-17, Rich Client SQL Server Memory Estimates

RAM per User		14.5 MB
Physical Memory (GB)	**Usable Memory (GB)**	**Estimated Number of Users**
4	2	140
8	6	420
12	10	700
16	14	980
24	22	1540
32	30	2110
48	46	3230
64	62	4360
96	94	6620

Managing memory

By default, SQL Server will allocate as much memory as is available on the machine. If there is a need to limit the memory available to SQL Server, see *Server Memory Server Configuration Options* in the *Microsoft Technet Library*.

Please also reference the following prior to adjusting memory:

- *http://support.microsoft.com/default.aspx?kbid=274750*
- *http://support.microsoft.com/default.aspx?scid=kb;en-us;811891*

To enable extended memory for 32 bit SQL Server use the following steps as a guideline.

- For systems with more than 4GB RAM, verify /pae is set in the boot.ini parameters.
- For systems with more than 16GB of RAM, verify that /3GB is **not** set in the boot.ini parameters if /pae is enabled.
- With SQL Server, enable awe memory:
  ```
  sp_configure 'awe enabled', 1
  RECONFIGURE
  GO
  ```
- Restart SQL Server Service

5.3.6.4 SQL Server Memory Pagefile Sizing for Teamcenter

Sizing benchmarks to 5,000 users required almost no pagefile space for SQL Server if there is sufficient RAM configured in the system.

5.3.6.5 SQL Server CPU Sizing for Teamcenter

For a typical APA 3000 user Rich Client scalability benchmark the SQL Server 2012 CPU utilization was about 12.1% peak, 5.1% average on a Dell 16 core Intel Xeon E5-2640 v3 CPU @ 2.6 GHz as Figure 5-14 shows. The SQL Server used for this benchmark is capable of a SPECint_rate2006 of 723, which equates to an SQL Server

Demand Rate of 36.9 at steady state for this Rich Client usage profile, or 0.012 per user (723 x 0.051 ÷ 3000), and 0.029 per user at peak (723 x 0.121 ÷ 3000).

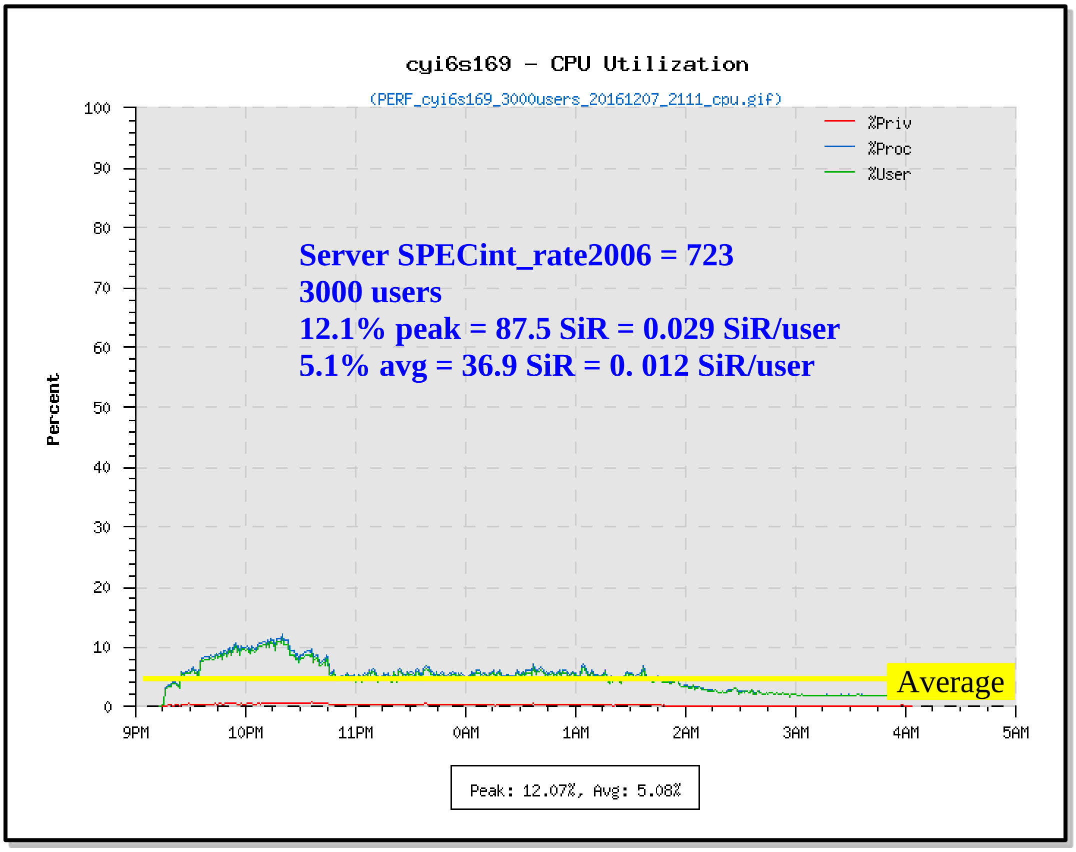


Figure 5-14, Rich Client SQL Server CPU Utilization Profile

In order to keep CPU utilization below 80%, a system with a SPECint_rate2000 rating of 110 is needed to serve SQL Server for this 3,000 user Rich Client usage profile.

This is somewhat more than the baseline release as seen in Figure 5-15. The blue portion of the columns indicates average CPU, the red portion peak CPU.

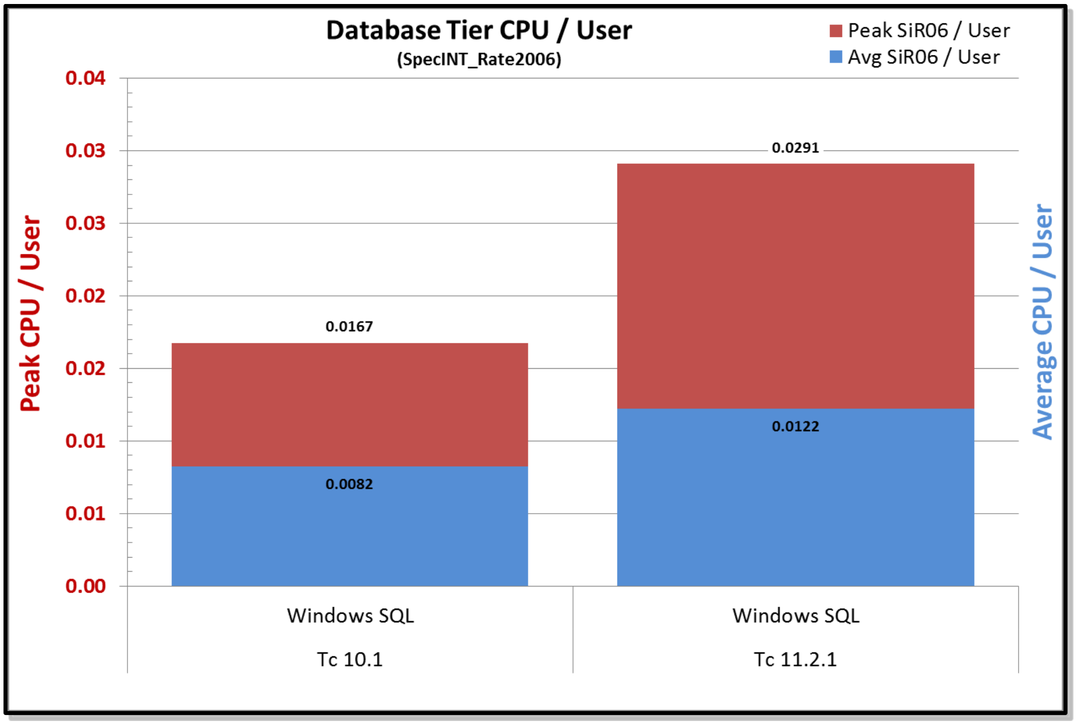


Figure 5-15, Rich Client SQL Server per User CPU Utilization – Rich Client

5.3.7 *Other Considerations*

5.3.7.1 BIOS settings

Processor Power States

Some processors support multiple power states intended to conserve energy when that is important. If performance is critical to your deployment, consider disabling C-States (processor states) and / or P-States (performance states) in the BIOS settings. Not all processors support these options, and not all BIOS implementations support configuring them. Refer to the hardware user's guide for your particular servers.

5.3.7.2 SQL Server High Availability

Although SQL Server Mirroring is not yet supported by Teamcenter, a number of other high availability options are available. See the whitepaper on *High Availability Options for Siemens Teamcenter on SQL Server* at the Microsoft Download Center, as well as other Teamcenter / SQL Server best practices, case studies, and other information.

5.3.8 *Other Considerations*

5.3.8.1 BIOS settings

Processor Power States

Some IBM processors support multiple power states intended to conserve energy when that is important. For example Power 7 processors support Automated Energy Management. If performance is critical to your deployment, consider disabling power management features. Refer to the appropriate IBM user’s guide for your particular system.

Chapter

6 Managing FMS & Volume Server Deployments

Go to main [Table of Contents]

Chapter

6 *Managing FMS & Volume Server Deployments*

This chapter focuses on deployment and sizing considerations of the Teamcenter File Management System (FMS). Use of Network Attached Storage (NAS) devices supporting NFS and CIFS protocols is supported with FMS.

NOTE

This section has not yet been updated for Teamcenter 11.2.1.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- Server installation for: *Windows*, *UNIX/Linux*
- *Using the File Management System*
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

6.1 Deployment Configurations

The File Management System (FMS) downloads and uploads file data from / to Teamcenter volumes for Teamcenter clients including the Rich, Thin, and Active Workspace clients, and CAD integrations. All Teamcenter file access / update is via FMS cache server processes, which read / write the files in volume servers, and stream the file(s) to/from clients as required. Multi-Site Collaboration also uses FMS servers to transfer data. TCFS is completely removed from Teamcenter components. For more information see the *Install a volume server* section under *Teamcenter server installation* in the *Installation on UNIX Servers* and *Installation on Windows Servers* manuals.

NOTES

An instance of the FMS server cache (FSC) must be installed and running on the Enterprise server with same user ID as the server pool manager for accessing volume data.

6.2 Volume Server Management Considerations

Teamcenter uses two types of volumes, standard volumes and transient volumes, both with an associated FMS process. Standard volumes are OS directories to store files managed by Teamcenter. Transient volumes are directories located on the Enterprise server that Teamcenter uses to store temporary data for transport between the Enterprise tier and clients in 4-tier deployments.

6.2.1 *Volume Backup*

Since the Teamcenter database contains references to content in the file vaults, volume files and database files should be backed up simultaneously. This will help ensure consistency between the database and the file vaults.

Cold backups are the most secure way to backup database and volume files together. This makes the system unavailable to users, and may not be an option for some deployments

Teamcenter provides a utility, backup_modes, to support pseudo hot backup. With this utility the administrator can place the Teamcenter volumes in one of three modes.

- Read-Only Mode: volumes are read-only state – no volume writes / deletes are allowed
- Blobby Volume Mode: writes to volumes are redirected temporarily to a specified location, allowing continuous availability, while the main volumes are backed up
- Normal Mode: the standard mode

For more information about the backup_modes utility, refer to the Teamcenter *System Administration Guide*.

6.2.1.1 Storage System Integration

Some storage system providers have incorporated rapid backup features that can be integrated with the backup_modes utility to accelerate and streamline the backup of Teamcenter database and volume files. For example, NetApp storage systems include SnapShot / SnapRestore features that can backup entire volumes in seconds. Most other storage providers include similar capabilities, although performance may vary. See the following URL for guidelines to integrate backup_modes with a NetApp system, which you should be able to adapt for other storage systems.

Backup and Recovery of Siemens PLM Teamcenter on NetApp Clustered Storage Solutions

Your storage system provider may also have deployment guidelines for their servers specifically when deployed with Teamcenter. For example, see the following URL for guidelines to deploy a NetApp system with Teamcenter.

Teamcenter Deployment and Implementation Guide on Data ONTAP Operating in Cluster-Mode

6.3 FMS/Volume Server Sizing

Sizing information is presented here separately for FMS servers (i.e. the root FSC cache server) and volume servers. The volume server can be the same machine where the FMS process runs, can utilize local direct attached disk, Network Attached Storage (NAS) servers, or Storage Area Network (SAN) systems.

NOTE

Both the environments and usage profiles used to simulate Teamcenter operations, including scenarios, transactions, number and types of files, operating systems and databases have changed since the previous Teamcenter Deployment Guide was published. ***For these reasons it is not valid to compare estimates in this revision of the guide with previous revisions.*** Use the simulation information in this chapter and extrapolate from that to estimate the target pilot environment.

6.3.1 *FMS Server Sizing*

NOTE

With the vast majority of today's high speed, high throughput, multi-core processors, two to four cores have the capacity to completely saturate a single gigabit network connection. Unless you anticipate extremely heavy file management traffic, a commodity 4-core system with gigabit network interface will likely suffice as your FMS server.

If you do expect significant numbers of file based operations, the following sections provide some guidelines to estimate server requirements.

Sizing the FMS server is highly dependent on the frequency of reads, writes, and deletes from from/to the Volume server and clients. For the most part this is directly related to the number of dataset exports and imports users perform. For this reason, it is difficult to map the number of named users to the number of file transfers that FMS must service.

Therefore FMS sizing information provided here is based on file transfer operations (exports, imports, deletes), not user counts. To help estimate the number of file operations in your production deployment, consider some important factors based on the following statistics:

- *Number of datasets opened* – This is determined by how many users are active and opening or loading datasets.
- *Number of datasets created and revised* – Dataset revisions result in a new dataset version being created, so revisions result in a create operation
- *Number of datasets deleted* – Datasets are deleted if a user actively deletes them or the dataset version limit is exceeded.

See also section 6.3.2.1 for information regarding estimating the amount of disk space required on the Volume server system.

6.3.1.1 FMS Server Sizing – Usage Profile

FMS server memory and CPU utilization were measured in the Teamcenter APA labs with a simulated multi-user load. FMS was deployed on a server machine accessing volumes from a separate network attached storage (NAS) system. Multiple processes were executed concurrently to simulate random, concurrent exports, imports, and deletes.

NOTE

Both the environment and usage profile used to simulate FMS operations, including number and types of files, has changed since the previous Teamcenter Deployment Guide was published. Further, this revision of the guide uses SPECint_rate2006 values for CPU estimates vs. the obsolete SPECint_rate2000 values in previous guides. *It is not valid to compare estimates in this revision of the guide with previous revisions.* Use the simulation information in this section and extrapolate that to estimate the target pilot environment.

For these measurements 250,000 exports (opens), 50,000 imports (creates), and 50,000 deletes were performed within a period of sixty minutes or less. Figure 6-1 shows the operations per minute simulated by this benchmark.

The number of operations per minute was randomized with a peak export rate of nearly 30,000 per minute.

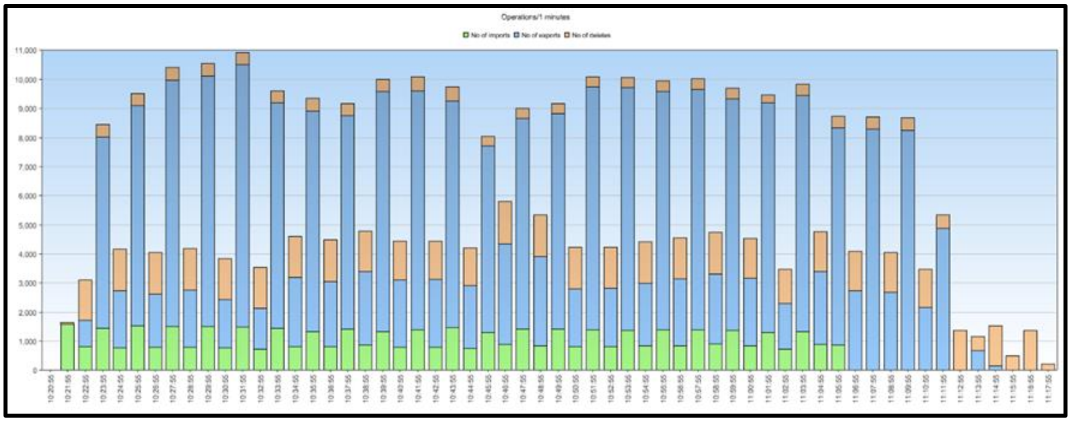


Figure 6-1, FMS Simulation – Operations per Minute

File sizes ranged from a few hundred bytes to several megabytes. Figure 6-2 shows the file size distribution of the 300,000 files used.

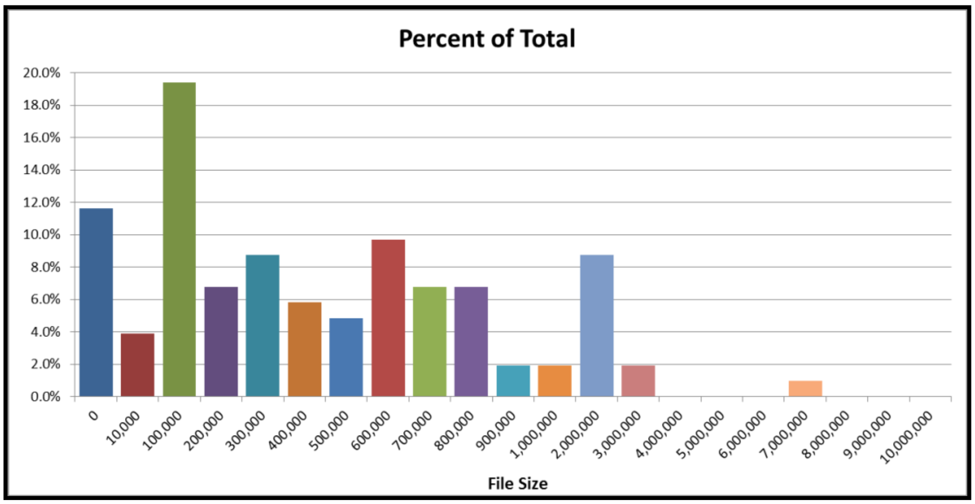


Figure 6-2, FMS Simulation – File Size Distribution

6.3.1.2 FMS Server CPU Sizing

The load profile shown in Figure 6-1 resulted in the following CPU utilization on an FMS server with two 3.07 GHz Intel® Core™ i3-540 processors and 8GB RAM (Figure 6-3).

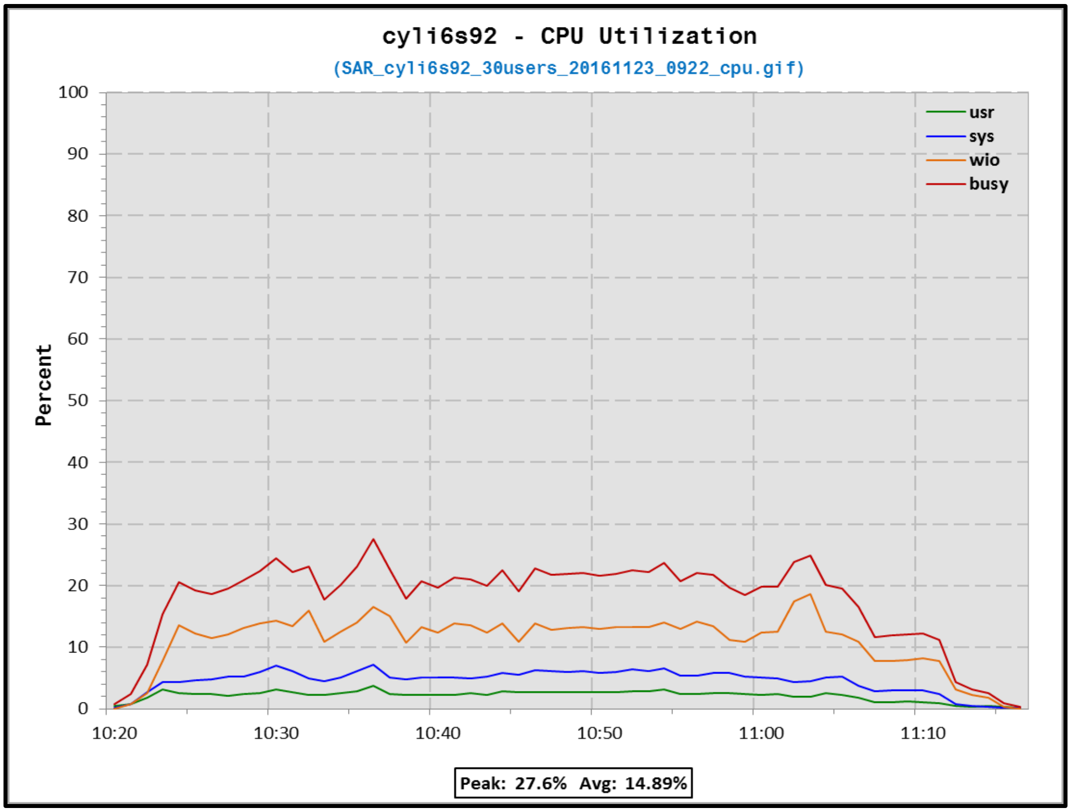


Figure 6-3, FMS Server CPU Utilization – Linux

Figure 6-3 shows that average CPU was less than 15% and peaked at 27.6%. The system used for these measurements (IBM x3250 M3, 2 x 3.1 GHz Intel® i3-540) is rated with a SPECint_rate2006 (SiR) of 61.8, yielding a SiR requirement of 26.0 at peak and 13.4 average.

To estimate your production system requirements, estimate how many file operations are expected compared to the load profile shown in Figure 6-1, then scale the CPU requirements accordingly.

6.3.1.3 FMS Server Memory Sizing

FMS uses relatively little memory, which is typically less than 1 GB.

6.3.1.4 FMS Server Swap Sizing

FMS also uses relatively little swap / pagefile, which is typically less than 1 GB.

6.3.2 *Volume Server Sizing*

Volume server sizing is primarily dependent on file sizes, frequency of access, anticipated numbers of revisions and versions maintained, all of which are different for each customer.

6.3.2.1 Volume Server Disk Space Sizing

Disk space requirements for the Teamcenter vault server are determined by the amount of data stored there. Vault server disk space can easily be estimated once you document the file characteristics of your business. To help estimate disk space, consider some important factors based on the following statistics:

- *Migrated Items* – Number of items expected to be initially managed by the Teamcenter system (that is, the number of items to be imported or migrated into the system).
- *Item Growth* – Number of new items/item revisions per month expected to be managed by the Teamcenter system
- *Item Revisions* – Average number of revisions expected for each item
- *File Size* – Average size of data files (Teamcenter datasets)
- *Dataset Versions* – Number of old dataset versions configured in Teamcenter (site preference AE_dataset_deafult_keep_limit).

Calculate the total space requirements and add 5 percent for file system structures. Also, add any requirements for operating system and application software. Calculations used to estimate required vault disk space in Appendix B, *Sample Usage Profiles*, assume a 125 KB file size, an average of four revisions, each with two versions.

When laying out volume definitions in the Teamcenter *Organization* application, try to organize multiple volumes, rather than a single large volume. Consider organizing volumes by user groups if everyone is in a single building on common LAN segment. Alternatively, you can organize volumes by location if users are in different buildings on isolated LAN segments.

6.3.2.2 Volume Server CPU Sizing

To estimate the required file server CPU capacity, characterize file access profiles to determine the NFS IOPS (Input/Outputs per Second) rate expected of the servers. Use the sample scenarios and load definition spreadsheet in appendixes A, *Sample User Scenarios* and B, *Sample Usage Profiles*, to help define your user's access profiles.

6.4 Volume Server Performance

Teamcenter volume server performance is impacted by the following factors:

- Network throughput
- Volume server capacity
- Volume server OS configuration

6.4.1 *Network Throughput*

Network throughput (i.e. bandwidth) presents the largest single impact to Teamcenter volume server performance. In most implementations, especially CAD implementations, Teamcenter must move large amounts of file data across the network to/from the volume server. Like all other Teamcenter components, the speed, latency, and congestion of your network all have an effect. For optimal volume server performance, the network between the volume server(s) and the FMS server(s) should be high speed, low latency, with minimal network errors. FMS accessing a volume server across a WAN results in very poor performance for file accesses; accessing across a satellite link is the worst possible scenario.

To minimize network related volume server performance issues:

- Place volumes directly on the FMS (root FSC) server. If volumes are located on Network Attached Storage (NAS), place the servers on the same LAN as the FMS server, ideally on the same subnet.
- Use a Gbit interface and network between the NAS server, the FMS server, and the network switches to which clients connect. This ensures that the combined traffic between multiple clients and the server does not congest the network path.
- Design the network topology with the shortest path possible between FMS/Volume servers and clients. This means keeping the number of switches and routers to a minimum.
- If users are separated from the main Teamcenter installation via WAN, consider deploying one or more remote FMS File Server Caches (FSCs) collocated with those users with their default volume set to the corresponding remote FSC. This will improve file load / update performance, while allowing access from all other geographic locations.

An excellent document with guidelines for Network Performance Tuning is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

6.4.2 *File Server Cache – Operating System TCP Resource Tuning*

By default, most operating systems do not set TCP parameters at optimum values for a system that manages as many individual transactions per second as the Teamcenter FMS server system. For example, heavy volume access traffic results in rapid use and release of sockets per second. If TCP is not tuned optimally, as in the case of TIME_WAIT setting, it can have a negative effect on performance. For example, if the default value of *TcpTimedWaitDelay* on the Windows operating system is left at 240 seconds, this can lead to socket exhaustion under peak loads by failing to release sockets in a timely manner. When this happens, requests to the TCFS server are delayed. An indication of this may be warning messages in the tcserver syslog files such as "`+++ You seem to have a slow IMAN-FS system`".

See also section 10.2.2.4, *Network Resources*, for further information about improving network performance.

In Windows deployments where FMS accesses the volume files over the network (i.e. CIFS shares), ensure volume file server is configured for SMB 2. SMB 2 significantly reduces the 'chattiness' of CIFS, improving file transfer performance. Windows Server 2003 / Vista and later support SMB 2. For example, on NetApp file servers running Data ONTAP 7.3.1 or later, be sure the server is configured as follows:

```
cifs.smb2.client.enable on
cifs.smb2.durable_handle.enable on
cifs.smb2.durable_handle.timeout 16m
cifs.smb2.enable on
cifs.smb2.signing.required off
```

Newer Data ONTAP versions may use different configuration parameters. Check your file servers' administration guides.

The TCP parameters in Table 6-1 were modified and/or applied on the FMS server on the Windows operating system in the APA lab. FMS Throughput tests show positive improvement in data processing rate with these settings. Similar TCP parameters are found in most UNIX implementations.

Table 6-1 FMS TCP Parameter Settings

Windows Registry Name **HKEY_LOCAL_ MACHINE\ SYSTEM\ CurrentControlSet\ Services**	**REG_DWORD Value Named**	**Default Value**	**Recommended Value**
Tcpip\Parameters\	TcpTimedWaitDelay	0x0000001e (240 secs)	0x0000001e (30 secs)
	MaxUserPort		0x00008000 (32768)
Tcpip\Parameters\ Interfaces	TcpAckFrequency		1
AFD\Parameters\	EnableDynamicBacklog		0x00000001
	MinimumDynamicBacklog		0x00000020
	MaximumDynamicBacklog		0x00001000
	DynamicBacklogGrowthDelta		0x00000010

6.4.3 *Volume Server Throughput Capacity*

Teamcenter volume servers are typically impacted first by the throughput of the network interface and then the machine's disk I/O capacity. Actual I/O requirements can vary considerably based on your usage patterns. The network interface cards in FMS/volume servers should be Gbit. Also, volume servers should utilize high throughput disk controllers and multiple, high-speed (rpm) disk drives. Consider virtual file systems such as RAID 0 to improve file transfer rates. Remember that some RAID configurations (for example, RAID 5) negatively impact file write performance. High throughput NAS systems, such as the NetApp 3100 series, provide very high throughput at very low access latencies, by supplementing the underlying RAID configuration with large read/write caches.

Also consider performance acceleration options for NAS / SAN file servers. For example, as Figure 6-4 shows, benchmarks of previous Teamcenter releases with NetApp's Flash Cache showed that inexpensive SATA drives performed reads 55% faster than more costly fiber channel disks without Flash Cache, although neither import nor delete performance was improved.

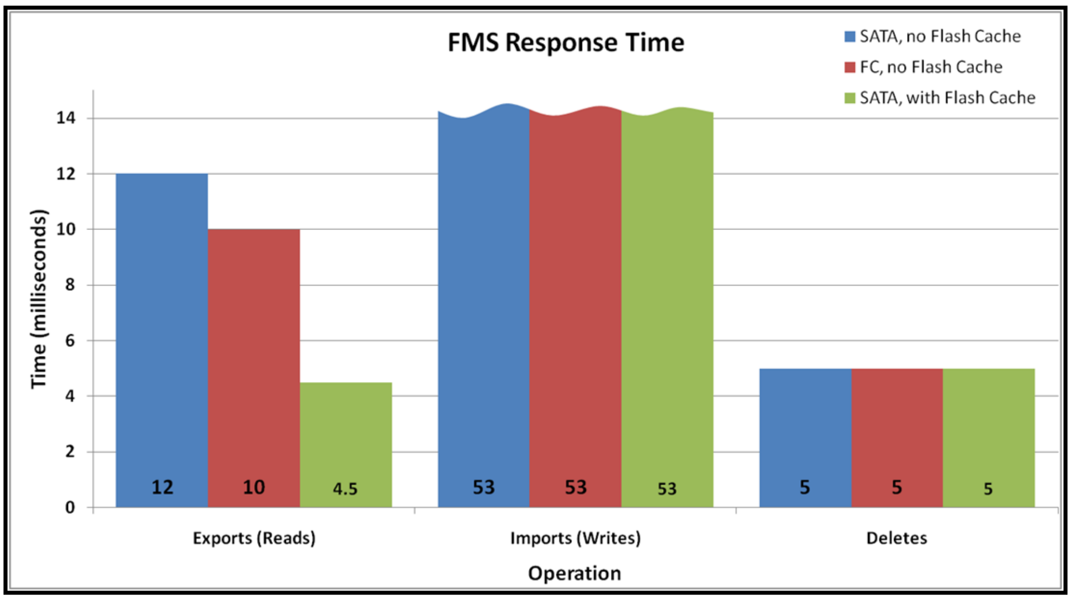


Figure 6-4, FMS Server Response Times with/without NetApp Flash Cache

Chapter

7 Managing Classic Multi-site Deployments

Go to main Table of Contents

Chapter

7 Managing Classic Multi-site Deployments

This chapter is intended to help you:

- Understand what to expect with regard to performance of typical Classic Multi-site operations.
- Understand the impact of various Classic Multi-site configuration and command line options.

NOTE

This section has not yet been updated for Teamcenter 11.2.1.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- Server installation for: *Windows*, *UNIX/Linux*
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

Teamcenter Classic Multi-site has a number of configuration and command line options that can have significant impact on the performance of the following operations:

- Multi-process ODS
- Object Synchronization
- Object Import/Export
- Publish & Unpublish
- Data_share

- DSA_Util

7.1 Classic Multi-site Performance Related Options

There are several options that can have an effect on Classic Multi-site performance and throughput.

7.1.1 *Object Synchronization Options*

Table 7-1, *Classic Multi-site Data Synchronization Estimates*, lists six Classic Multi-site scenarios measured at various bandwidth and latency delays with the resulting time per item to synchronize data to a single remote site. The following times are derived from these benchmarks.

Table 7-1, Classic Multi-site Data Synchronization Estimates

Scenario	Size of Assembly	Data Synchronization Environment	Data Sync Rate
1	3083 component assembly with 219 unique datasets	• data_share to one site • LAN • IDSM - 4 x Quad-Core AMD Opteron(tm) Processor 2350 @ 2 GHz, 2 GB RAM, 8 GB Swap • Microsoft Windows Server 2012 R2 Standard	2.1 sec/item
2	3083 component assembly with 219 unique datasets	• data_share to one site • 100ms latency/T1 Bandwidth • IDSM - 4 x Quad-Core AMD Opteron(tm) Processor 2350 @ 2 GHz, 2 GB RAM, 8 GB Swap • Microsoft Windows Server 2012 R2 Standard	15.5 sec/item

Table 7-1, Classic Multi-site Data Synchronization Estimates (cont.)

Scenario	Size of Assembly	Data Synchronization Environment	Data Sync Rate
3	3083 component assembly with 219 unique datasets	• data_share to one site • 300ms/T1 • IDSM - 4 x Quad-Core AMD Opteron(tm) Processor 2350 @ 2 GHz, 2 GB RAM, 8 GB Swap • Microsoft Windows Server 2012 R2 Standard	18.6 sec/item
4	654 item assembly with 67 unique datasets	• data_share to one site • LAN • IDSM - 4 x Quad-Core AMD Opteron(tm) Processor 2350 @ 2 GHz, 2 GB RAM, 8 GB Swap • Microsoft Windows Server 2012 R2 Standard	1.9 sec/item
5	654 item assembly with 67 unique datasets	• data_share to one site • 100 ms/T1 • IDSM - 4 x Quad-Core AMD Opteron(tm) Processor 2350 @ 2 GHz, 2 GB RAM, 8 GB Swap • Microsoft Windows Server 2012 R2 Standard	10.4 sec/item

Table 7-1, Classic Multi-site Data Synchronization Estimates (cont.)

Scenario	Size of Assembly	Data Synchronization Environment	Data Sync Rate
6	654 item assembly with 67 unique datasets	• data_share to one site • 300 ms/T1 • IDSM - 4 x Quad-Core AMD Opteron(tm) Processor 2350 @ 2 GHz, 2 GB RAM, 8 GB Swap • Microsoft Windows Server 2012 R2 Standard	14.0 sec/item

7.1.2 *Object Import/Export Options*

A wide variety of factors influence the length of time required to transfer data between sites using remote import. Since the remote import action is essentially composed of three operations (export – data transfer – import) the characteristics of the data and the options selected during the remote import action have as much impact on the elapsed time of the transfer as the profile of the environment which has been configured.

The chart in Figure 7-1 is a comparison of the time required to transfer data using FTP and Teamcenter Multi-site. The same 1GB and 2GB datasets which were exported from Teamcenter were used in the measurement. The results show that data transfer using Multi-site is as efficient as FTP at lower latencies and slightly faster than FTP when compared in a WAN environment with higher latency values.

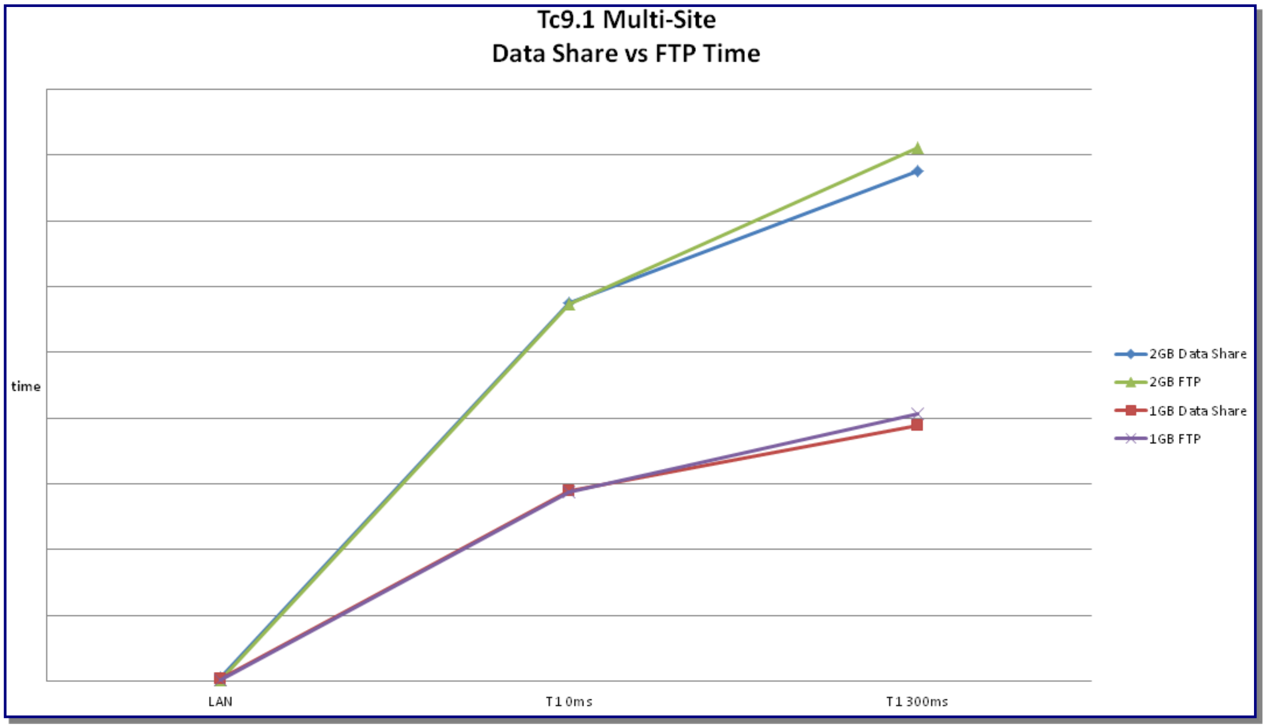


Figure 7-1, Multisite – FTP Comparison

7.1.3 *Transfer Ownership Options*

Measurements taken in the Teamcenter Performance and Scalability Labs illustrate the difference between transferring an assembly with the *"Transfer of Ownership* "option selected vs. the default "*Replica"* option. Table 7-2 shows that for the assemblies measured, transferring ownership can take 15-25% more time than simply replicating the assembly. Larger or different sets of data may take more or less time.

Table 7-2, Classic Multi-site Remote Import – Transfer Ownership vs. Replica

Size of Assembly	Network latency/bandwidth	Increase in Elapsed Time for *Transfer Ownership* vs. *Replica*
87 Component Assembly	0.0 ms/T1	23%
87 Component Assembly	100 ms/T1	19%
87 Component Assembly	300 ms/T1	17%
3083 Component Assembly	0.0 ms/T1	15%
3083 Component Assembly	100 ms/T1	15%
3083 Component Assembly	300 ms/T1	14%

Methods of transferring data between sites in a Multi-site configuration include the following:

- Data Share with Transfer Ownership (via command line)
- Data Share with default Replicas only (via command line)
- Rich Client Export with default Replicas only

Measurements were performed transferring a 3083 BOM line NX assembly over LAN and T1 (1,544 kbit/sec) bandwidth lines with latencies on the T1 line of 0ms, 100ms, 200ms & 300ms. The results of the measurements are illustrated in Figure 7-2 below.

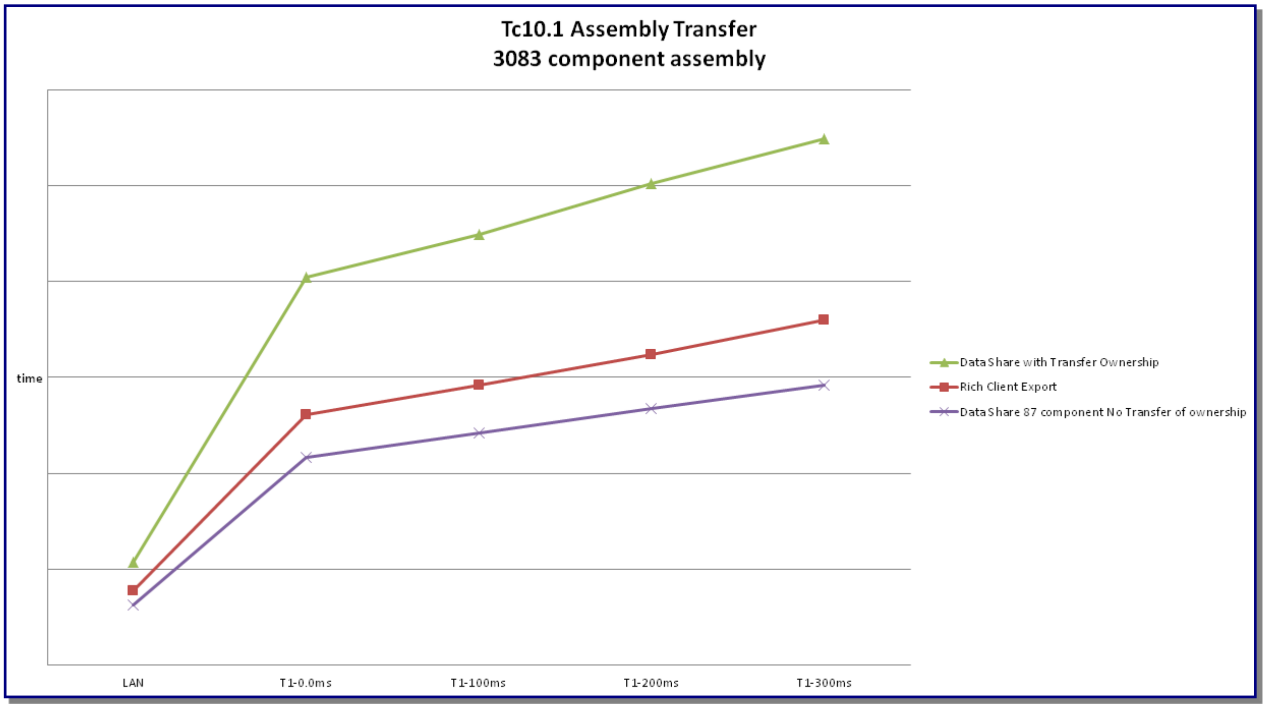


Figure 7-2, Large Assembly Transfer WAN Characteristics

The results indicate that the transfer of ownership option has a larger negative impact on elapsed time than an increase in latency on the network.

Measurements taken with a smaller assembly at a slower bandwidth and higher latency also indicate a trend of additional time to transfer ownership as indicated in Figure 7-3 below.

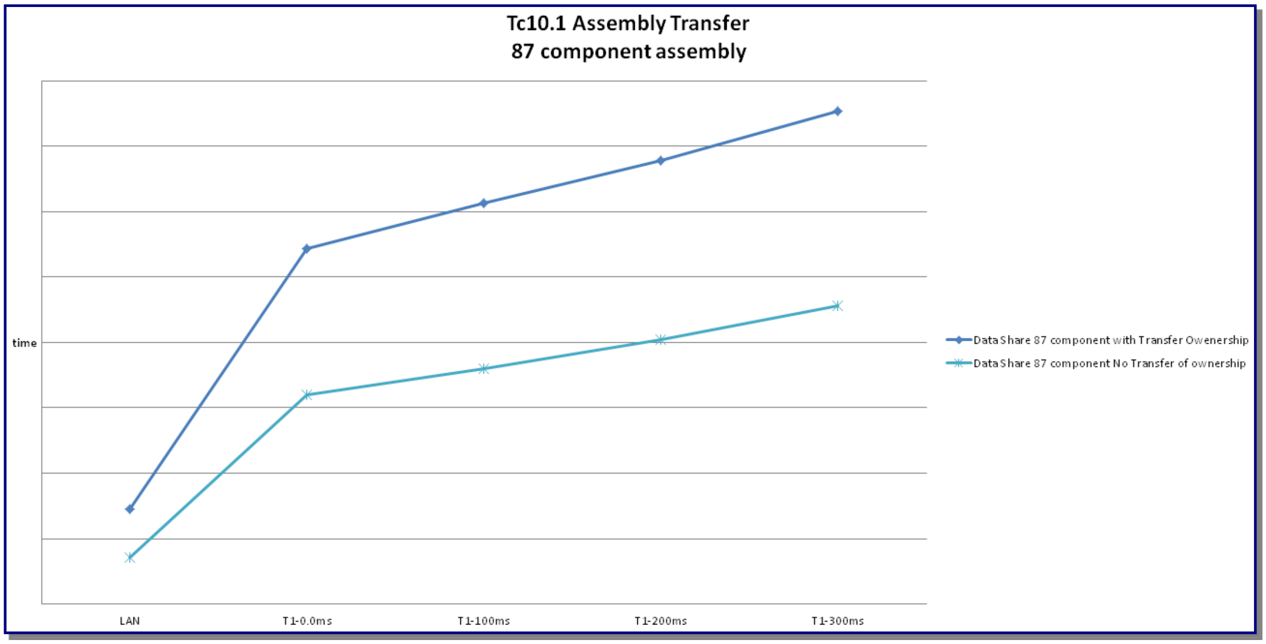


Figure 7-3, Small Assembly Transfer WAN Characteristics

7.2 Publish & Unpublish Performance

Publishing and unpublishing objects involves the transfer of *'publication'* records from the site to the system on which ODS is executing. Since publication records are fairly small, increases in network latency do not impact these operations as much as other Multi-site functions. End-user response times for publishing single Items at increasing network latency values have been shown to be very consistent (Table 7-3).

Table 7-3, Classic Multi-site Publish % LAN vs WAN using Rich Client

Number of Items Published to Single Default ODS	Network Characteristics (latency/bandwidth)	Increase in Latency	Increase in Time to Publish
1 item	0.0 ms/T1	1x	1.15x
1 item	100 ms/T1	100x	1.16x
1 item	300 ms/T1	300x	1.63x
87 items	0.0 ms/T1	1x	1.06x
87 items	100 ms/T1	100x	2.45x
87 items	300 ms/T1	300x	7.22x
219 items	0.0 ms/T1	1x	1.09x
219 items	100 ms/T1	100x	2.57x
219 items	300 ms/T1	300x	7.50x

Latency has very little impact on the time required to transfer metadata-only structures or data in Teamcenter Multi-site. As latency in the environment increases, the increase in time required to transfer the metadata only Teamcenter objects remains fairly constant.

Measurements using a 5,000 BOM line structure and transferring a Folder containing 400 Items are shown in Figure 7-4 below.

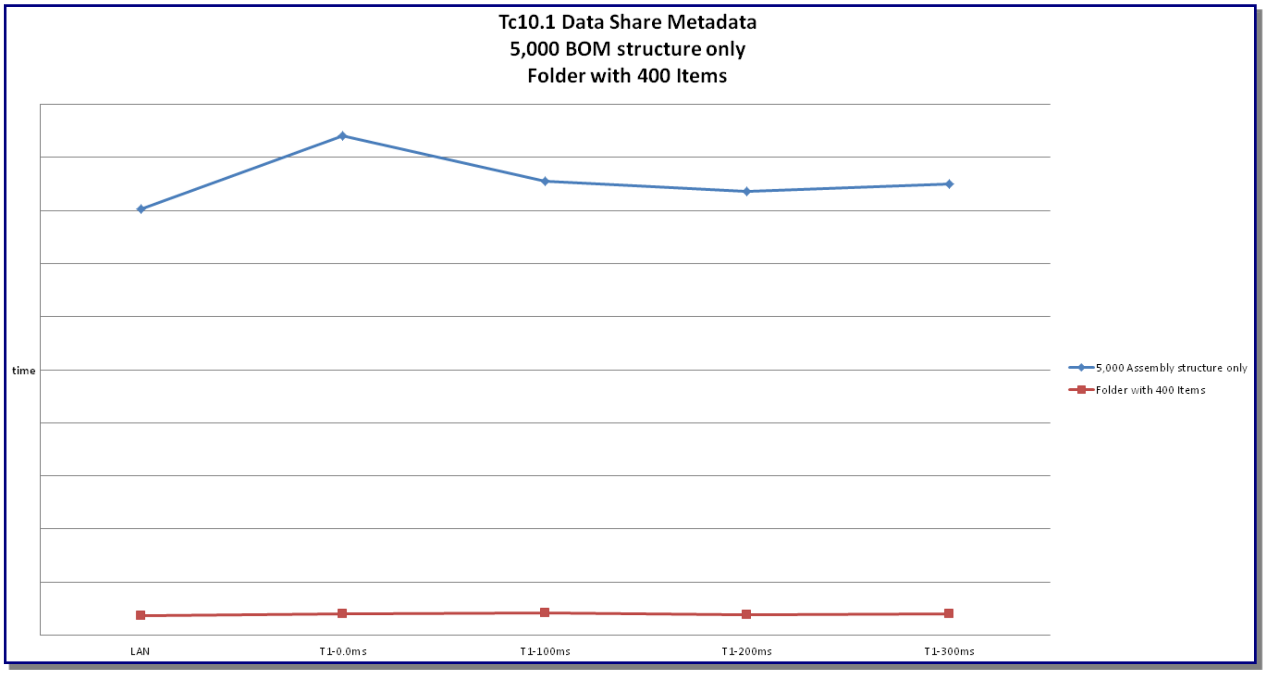


Figure 7-4, Data Share WAN Characteristics

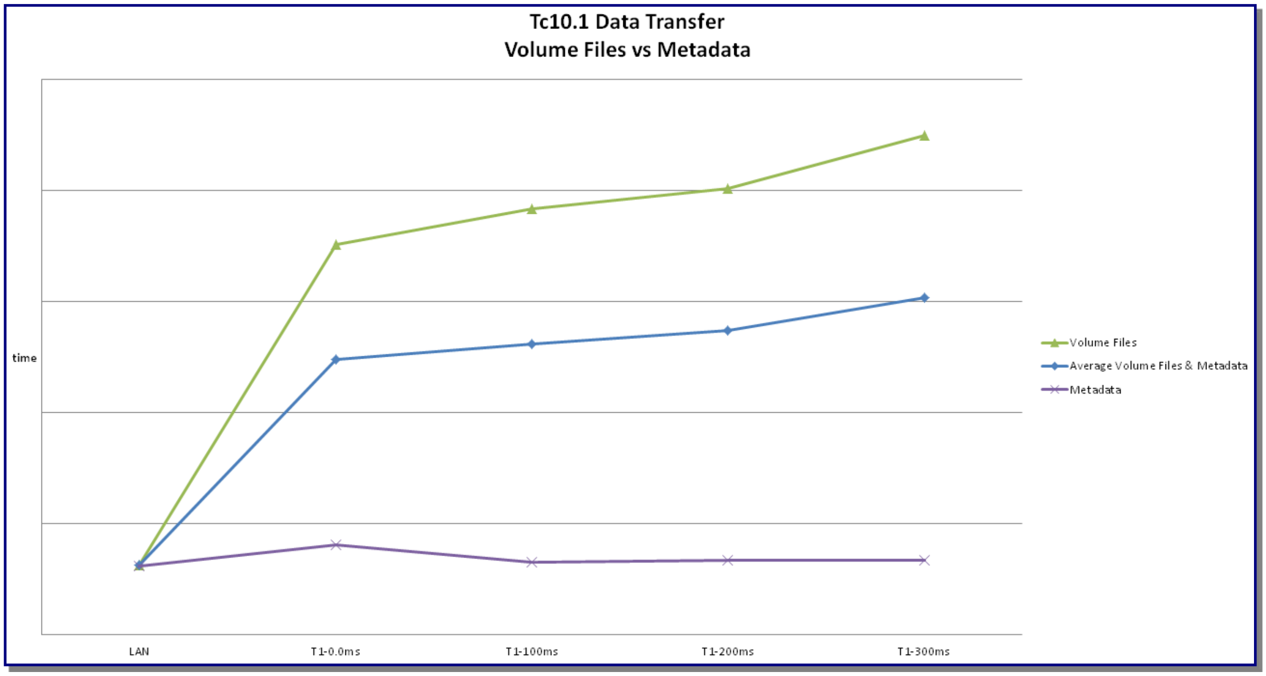


Figure 7-5, Data Transfer WAN Characteristics – Files vs. Metadata

Multi-site measurements of Data Transfer times (Figure 7-5) demonstrate bandwidth and network latency has very little detrimental impact on the time required to transfer metadata only sets of data. Sets of data which contain physical volume files are much more susceptible to the overhead of increased network latency and reduced bandwidth.

Figure 7-6 below illustrates the effect of increased latency and decreased network bandwidth on larger and smaller datasets being shared between sites. In this case, the difference in bandwidth from running in a LAN environment to a T1 environment has a much larger effect on transfer times than an increase in latency.

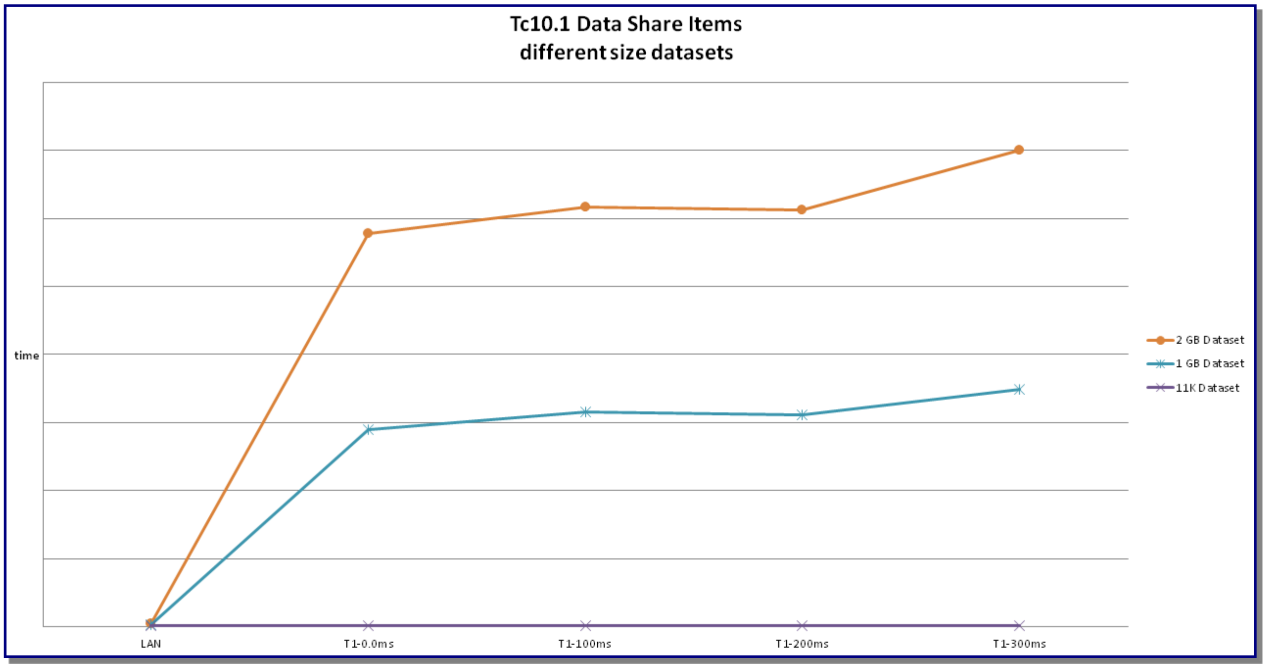


Figure 7-6, Data Share WAN Characteristics – Different Data Sizes

NOTE

Publication of a single item to the ODS enables the entire assembly represented by that item to be imported at remote sites (assuming that all correct permissions have been established for such an action). Specific items within the assembly will not be visible to users performing remote searches if only the top level item is published. In order for remote users to have the ability to search for all of the items in an assembly, all of the items in the assembly must be individually published.

If using command line Data_Share, no Items need to be published to push/pull the assembly if all ACLs are set up correctly. Once the assembly has been shared all Items are able to be remotely found with a simple search.

7.3 Search Performance

To improve the performance of ODS when potentially large number of users may be searching it, consider implementing Multi-process ODS.

Using Multi-process ODS is fully documented in the Teamcenter *Online Help Library* under *Sharing Data*. Search for *ODS Configuration*.

7.4 Network Planning for Classic Multi-site

An in-depth analysis of expected user activities is an important step that must be performed as a pre-cursor to network and site planning for a Classic Multi-site deployment. The frequency and types of activities that will be performed by end users will be dramatically impacted by the configuration of the Teamcenter Classic Multi-site environment.

The proportion of searches for published data versus actual data transfers (via remote imports or data synchronization) will determine the optimum distribution of resources (memory, processing, and network resources). The relative performance of remote searches will be a function of the processing speed and memory resources of the node on which the Object Directory Service (ODS) is running. The performance of data transmission between sites in a Classic Multi-site Teamcenter environment is directly proportional to the processing and memory resources of the node(s) on which the Integrated Distributed Services Manager (IDSM) is running and the latency and bandwidth of the network connecting the sites.

An excellent document with guidelines for Network Performance Tuning is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

7.4.1 *Estimating Classic Multi-site Network Requirements*

By estimating a number of operational factors that contribute to the time required to keep systems in a Classic Multi-site environment synchronized, you can determine how much network capacity will be required for your specific deployment. Factors that should be considered include:

1) Number of users creating/revising data
2) Number of Item changes per user per synchronization period
3) Number of Items created per user per synchronization period
4) Synchronization period – how often are synchronizations performed
5) Synchronization window – how many hours per sync period will data transfer be allowed
6) Number of Item Revisions changed per sync period (#1 times #2)
7) Number of New Item Creations per sync period (#1 times #3)
8) Average number of Item Revisions per Item

9) Number of Item Revisions synced per Item – will all revisions be synchronized or just the latest?

10) Number of datasets under the Item Revision. For those datasets that will be synced:

    - Average NX dataset file size (KB)
    - Average TC Vis dataset file size (KB)
    - Average file size of other datasets (KB)

Estimate each of these factors for your Classic Multi-site deployment to derive:

11) Kbytes transferred (synced) per period

    (sum of dataset file size) x (new item creations + (item revs x item revs synced))

    (#10) x (#7 + (#6 x #9))

12) Min Bandwidth Required to meet Sync period window (Kbits/s)

    (Kbytes transferred per Synchronization period x 8) ÷ (Synchronization window in seconds)

    (#11 x 8) x (#5 x 3600))

13) Recommended Bandwidth Required to meet Sync period window (Kbits/s)

    (Min Bandwidth Required x 110%)

From these you can estimate the network bandwidth requirements for the synchronization activity you expect. Be certain to set the IDSM_ft_buffer_kb environment variable appropriately for the bandwidth and latency characteristics of each network link in your Classic Multi-site environment according to instructions in the *Teamcenter Help Library*; search for *IDSM_ft_buffer_kb*.

7.5 Classic Multi-site Collaboration Firewall Considerations

Classic Multi-site Collaboration provides two methods for communicating through firewalls. You can configure a site to communicate using the HTTP or HTTPS protocol, or remote procedure call (RPC). Classic Multi-site processes, specifically IDSM and ODS, use RPCs by default and therefore are not Firewall friendly as initially installed.

Using Classic Multi-site with Firewalls is fully documented in the *Teamcenter Help Library*; search for *Using Multi-site Collaboration through a firewall*. This section documents configuration of ODS and IDSM for use with a Firewall, e.g. how to use named sockets, etc.

Classic Multi-site also supports a HUB configuration for use in a ‘DMZ’ to isolate external Classic Multi-site users from the main Teamcenter site. Find detailed information for hubs in the *Teamcenter Help Library*; search for *Multi-site Collaboration network topology*.

See also section **3.8**, *Teamcenter and Firewalls*, for additional information about using firewalls with Teamcenter Classic Multi-site Collaboration.

7.6 Classic Multi-site Collaboration Proxy Server Considerations

Since FMS was introduced with Teamcenter Engineering 2005 Multi-site has transferred files via FMS. Starting with Teamcenter 2007.1 FMS is also used to transfer the metadata. The 2007.1 onward methods will only be discussed here.

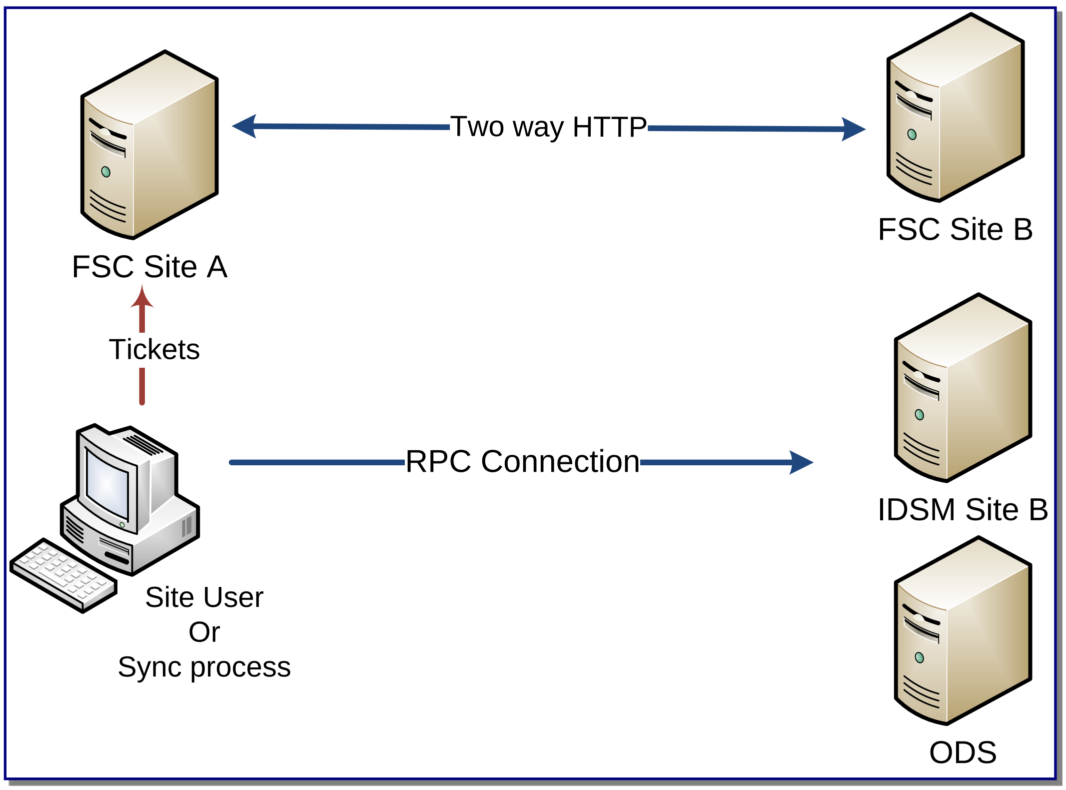


Figure 7-7, RPC Multi-site and FMS

The sequence is approximately as follows:

User or utility pulls data: The first rule of FMS is that it will always pull data; the only exception to this is when a client saves data.

1) The initial request is to the ODS via RPC to locate the required objects.
2) Once the objects required are determined and the user requests a transfer operation (Tools-Import-Remote), a connection from the utility or the tcserver is made to the remote IDSM process.

3) The IDSM exports the metadata required into a file it creates in its transient volume.
4) The FMS tickets for the metadata are returned to the requesting site.
5) The tcserver or utility sends the ticket to its FSC.
6) The FSC will either connect directly to the remote FSC or routes the request via the ExitFSC.
7) The data is pulled from the remote site
8) Once the metadata has been imported, a POST request is performed (actually a roll back is sent to the remote site to delete the metadata from the transient volume.

Next the file data is requested in much the same way. The IDSM is requested to generate the tickets for the files requested, the ticket for the file holding all tickets is returned and the receiving tcserver or utility uses sends the ticket to the FSC to transfer the data. Unlike the metadata process the file transfer is accomplished by the FMS system without use of the tcserver or utility. Control returns to the tcserver or utility once the file transfer is complete, again a POST operation is used to tell the remote site to clean up its transient volume data. This cycle is repeated as many times as required to move all the requested data.

User or utility pushes data: This time the user selects data to be sent to the remote site, as mentioned early FMS always pulls data so tickets are sent to the remote site so data can be pulled.

9) The tcserver or utility process exports the metadata and writes it into a file held in the transient volume.
10) The ticket for the file is sent via the RPC connection to the remote site, which uses its FMS system to pull the metadata data into the local FMS system and imports the data.
11) On completion an FMS POST is sent to delete the data from the transient volume.
12) The local tcserver or utility now creates a file of tickets for the required data files.
13) Again the ticket for that file is sent to the remote site via RPC and the remote site uses FMS to pull the data deleting the file on completion.

Confusion often arises from the fact FMS pulls the data even when the operation is a data push.

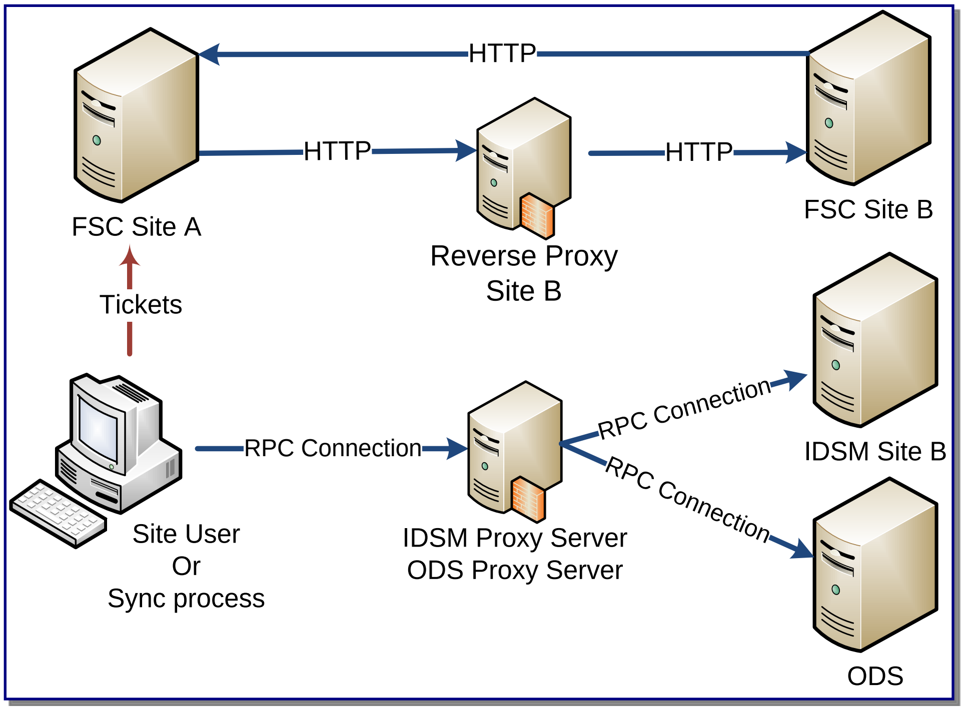


Figure 7-8, Multi-site Proxies

RPC Multi-site provides its own Proxy system which in older Teamcenter versions all RPC versions provided support for both metadata and files. From Teamcenter 2005 onwards you will require a standard http proxy for the FMS traffic to get equivalent functionality (Figure 7-8). The remote site will use the proxy server name in its definition of the remote site FSC.

Both Multi-site RPC and http proxy server can be daisy chained i.e. one proxy can forward to another proxy. This allows corporate access proxies to be defined on each continent if required.

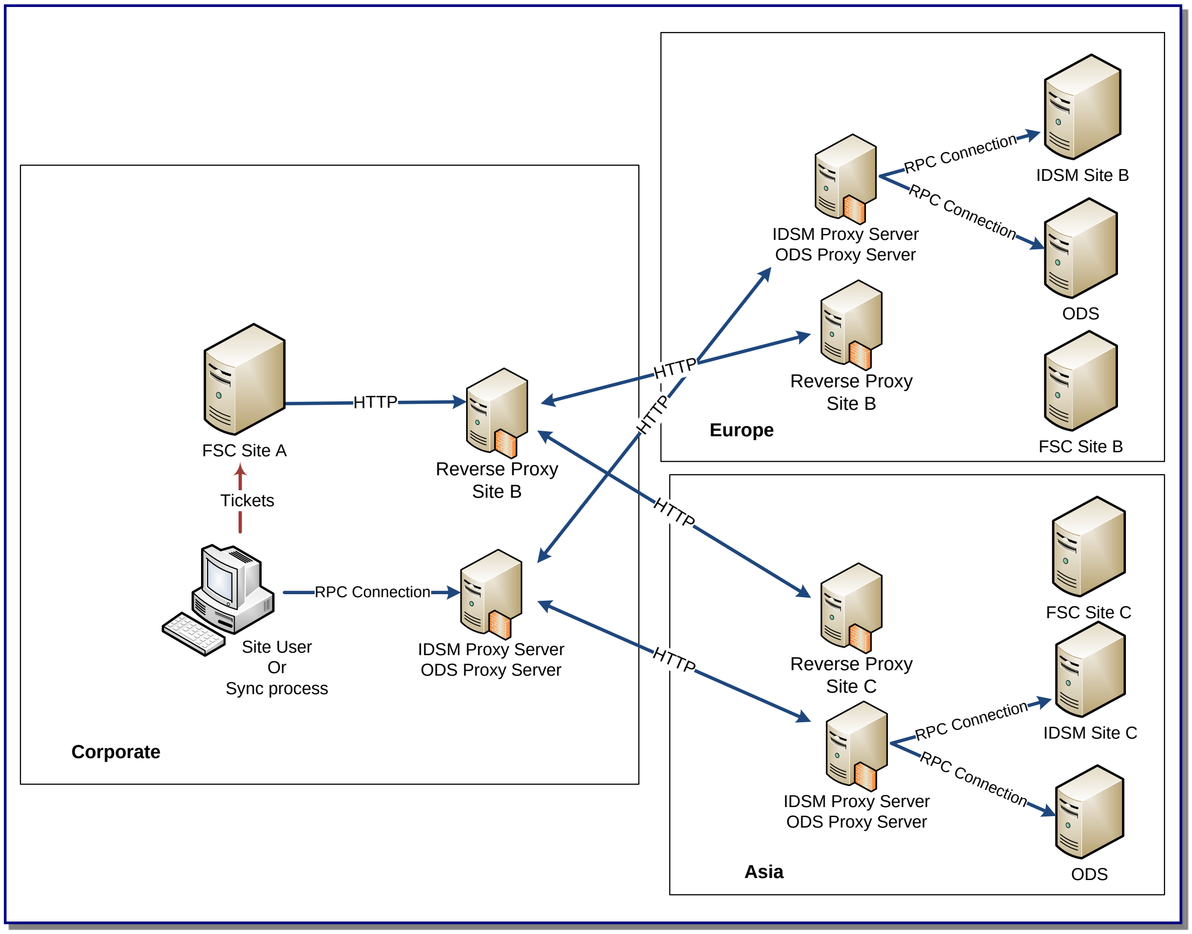


Figure 7-9, Daisy Chained Proxies

7.7 Classic Multi-site Collaboration using HTTP or HTTPS protocol

The requirement to install ODS/IDSM for Multi-site using the RPC protocol is not needed if HTTP or HTTPS protocol is used. When HTTP/HTTPS protocol is used, the Server Pool process replaces ODS/IDSM processes. A Single Sign-on (SSO) installation is required for the HTTP/HTTPS protocol to be used. To install SSO a Lightweight Directory Access Protocol (LDAP) installation must already exist in the environment. Both RPC and HTTP/HTTPS protocols can be used simultaneously in a Multi-site confederation.

Apache LDAP was installed in the APA lab for benchmarking purposes with Multi-site using both HTTP and SSO. The LDAP installation was not intended for more than the purpose of benchmarks for Multi-site. An LDAP installation for an entire company would be much more involved. Tables 7-4 and 7-5 indicate the required software components for different Multi-site protocols.

Environment Configuration Requirements for HTTP/HTTPS vs. RPC Protocols:

Table 7-4, HTTP/HTTPS vs. RPC Protocol

HTTP/HTTPS	Notes
1. Teamcenter 4-tier Installation is required	1. Web Server deployment is required to use HTTP/HTTPS protocol 2. The Server Pool tier assumes the role of the IDSM/ODS
2. Lightweight Directory Access Protocol (LDAP)	1. Some form of LDAP (i.e. Windows Active Directory) is required to be installed
3. Single Sign On "SSO" module is required	1. SSO must be installed in both %tc_root% and on each Rich Application Client (RAC)
4. tcssoid and tcssols war files deployed to the web server using insweb.	
5. Certificates required for HTTPS	
RPC Protocol	**Notes**
1. Teamcenter 2 or 4 Tier Configuration	
2. ODS & IDSM	1. IDSM must be installed at each site in the confederation 2. Minimum of one ODS required in the Multi-site confederation

Chapter

9 *Managing Teamcenter Cloud Deployments*

Go to main Table of Contents

Chapter

9 Managing Teamcenter Cloud Deployments

This chapter is intended to help you:

- Understand basic Cloud concepts as they pertain to Teamcenter deployments
- Understand the impact of various Teamcenter configuration options in the Cloud
- Estimate capacity requirements for CPU, RAM, and I/O

This chapter has not been updated from the previous Teamcenter release.

NOTE

This section has not yet been updated for Teamcenter 11.2.1.

NOTE

For additional information about the topics covered in this chapter, see the following references:

- Teamcenter *What's New in Teamcenter*
- Teamcenter *Support and Certification Announcements*
- Server installation for: *Windows*, *UNIX/Linux*
- The various *System Administration* guides

These and others can be found in the *Teamcenter installation and administration support* page on GTAC support.

9.1 Cloud Overview

Commercial cloud environments can provide an alternative to the purchase / lease and consequent operational costs of hosting computing / networking equipment in your facility (i.e. on your premises). Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with nominal management effort or service provider interaction[65].

There are often deployment options available from providers for the location of the cloud infrastructure. Purely public clouds host everything in the providers' facilities (off-premise) with very high levels of availability and disaster recovery capabilities, and are accessed over the public internet. Private clouds are also generally hosted off-premise with similar availability and disaster recovery capabilities, but are accessed over private networks and may be on a dedicated infrastructure (not shared with other customers). Some providers will offer combinations of deployment models, and many offer multiple Operating System options.

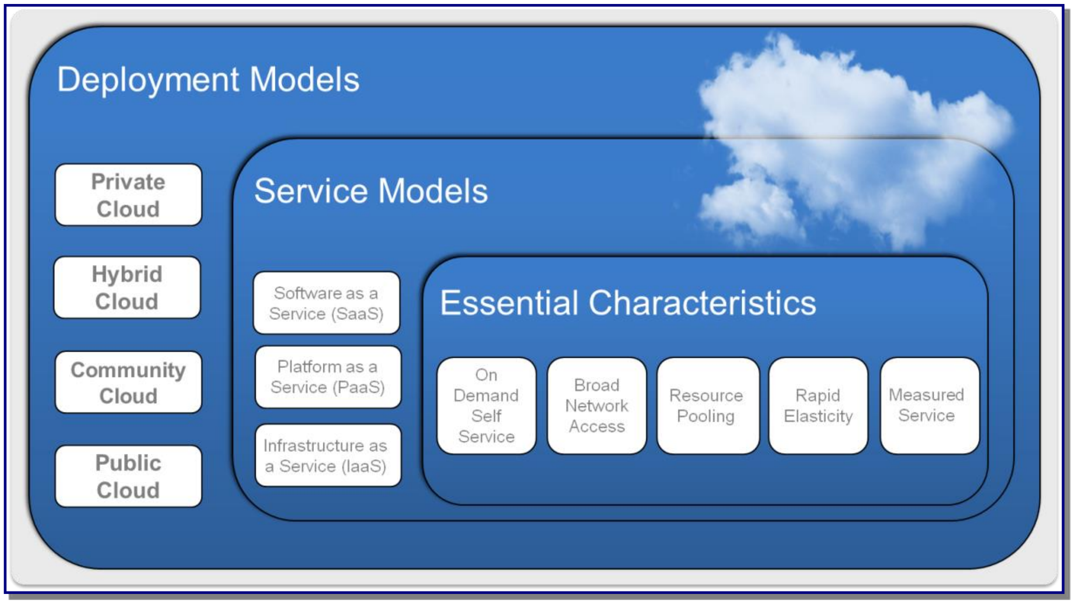


Figure 9-1, Sample Cloud Deployment Models

Depending on the provider, three levels of service are typically offered – Infrastructure, Platform, and/or Software.

The field of cloud providers and the level of services they offer are expanding rapidly, and not all providers or service levels are supported by Siemens PLM Software. Also, the operating system(s) and version(s) offered by the provider must be supported by Siemen PLM Software. For this release of Teamcenter, only Infrastructure as a Service (IAAS) is supported.

Future Teamcenter releases may support additional providers, or additional services levels, so this document may be out of date. For a list of supported cloud providers and levels, operating systems, and other third party software, always consult the *Operating*

[65] National Institute of Standards and Technology (NIST – see http://www.nist.gov/itl/cloud/)

System and 3rd party Certifications pages for the relevant Teamcenter release on the PLM Global Technical Access Center (GTAC) site.

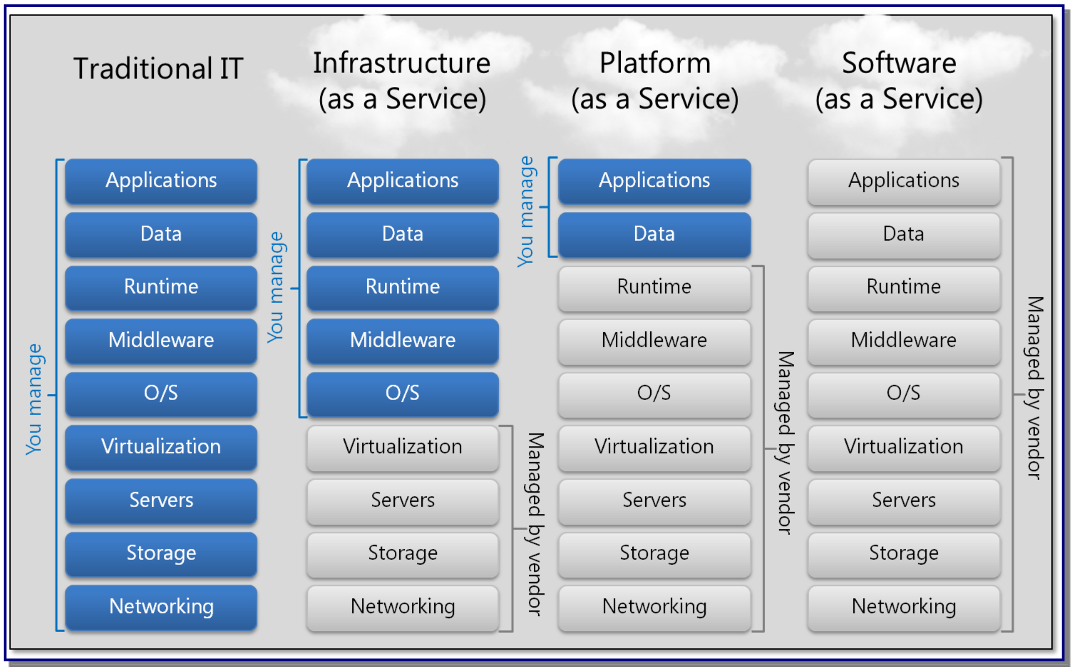


Figure 9-2, Cloud Service Levels

9.2 Cloud Deployment Overview

Deploying Teamcenter in cloud deployments is similar to on-premise deployments in many respects. Some cloud characteristics, which vary among providers, require special consideration. These will be noted in the relevant provider specific section of this chapter. Although Siemens PLM has assessed only those cloud providers that are or will be certified for Teamcenter, these characteristics may also be applicable to other cloud providers.

Creation, administration, and support of the underlying cloud servers is beyond the scope of this document, just as administration and support of on-premise servers would be. Consult your cloud provider for instructions and deployment considerations for cloud resident servers.

NOTE

Cloud providers may use various names to refer to a single computing node. Computing nodes may be a standalone server, or a virtual machine (VM) that shares hardware with other VMs. In most cases computing nodes are deployed as a VM regardless of the underlying hardware. Therefore the term server will be used in this document to a single computing node.

9.2.1 *Common Considerations for Teamcenter in the Cloud*

Prior to deploying Teamcenter in any cloud, you will need to:

- Contract with a cloud provider for IAAS access
- Create one or more billing accounts
- Create server instances with the desired operating system(s), associate storage, and network resources

Once the required servers are online with appropriate resources assigned to them, install and configure Teamcenter as you would for an on-premise deployment with the following considerations.

In general, the following apply to all cloud deployments:

- Teamcenter is currently supported only in clouds offered as an *Infrastructure As A Service* (IAAS).
- Unless deployed in a private cloud over a secure, dedicated network link, it is very highly recommended that all Teamcenter cloud deployments are configured with Secure Sockets Layer (SSL) enabled over HTTPS. Use a properly registered private key and certificate registered with a trusted authority. Not doing so puts your intellectual property at risk of being compromised on the public internet.
- Just as with on-premise deployments, Teamcenter components may be deployed on one or on multiple servers.
- If available from your Cloud provided, consider creating a Virtual Private Cloud (VPC) using the connectivity option of Network Address Translation (private subnets). This enables you to create a virtual network topology which connects your existing infrastructure to a set of isolated Cloud compute resources, including subnets, via a virtual private network (VPN) connection. With many Cloud providers, you can extend your existing management capabilities and security services such as DNS, LDAP, Active Directory, firewalls, and intrusion detection systems to include your Cloud resources, and protect the information there the same way you do now.
- Response times will generally be less consistent compared to an on-premise deployment due to 1) variation of the public WAN connections and 2) variation induced by the nature of the VMs on which the cloud service is based. Standard deviations of response times were found to be two to three times that of an isolated on-premise deployment.
- If Teamcenter is deployed on multiple servers, care should be taken to configure Teamcenter components to use the server hostname only when referencing other servers in the same cloud, and not the Fully Qualified Domain Name (FQDN). Use of FQDNs may cause network traffic to pass through the cloud firewall, outside the cloud, resulting in poor server response times. Preplan your VM names with a repeatable pattern such as “tcpoola”, “tcpoolb” to simplify

administration. In some cases, it may be necessary to use IP addresses rather than host names to achieve the desired behavior.

- If migrating an existing Teamcenter deployment to a cloud, consult with your provider about data migration options. Transferring large amounts of database and/or volume data via the internet will be very slow, potentially taking days or weeks. Some cloud providers offer a transfer service that accepts portable disk media and copies the data thereon to a cloud resident storage area.
- Since most cloud resident servers share a common I/O infrastructure for disk and network, I/O performance can fluctuate considerably. This can cause appreciable variation in server performance, especially the database server. Consult with your provider about I/O options that provide higher throughput, lower latency disk access, and more consistent performance.
- Be certain that you understand the shutdown / reboot behavior of all cloud servers on which Teamcenter is deployed. Most cloud compute servers will lose some memory & local disk storage (i.e. instance storage) if the server is shut down for any reason, planned or not. Some programs or settings may be lost between reboots, depending on the cloud provider and server type. Taking a server completely offline may lose all VM configuration data, depending on your cloud provider.
- Be certain the costing model is understood and mapped to your typical usage patterns prior to contracting for cloud services. CPU, RAM, and storage costs are usually part of the costing model, but some providers also charge for disk and/or network I/O, ingress or egress[66], or other resources used.
- Be sure to track costs closely during the early period of your deployment. Even the most careful initial estimates / pilots may prove inaccurate for production utilization.
- Consider the geographical nature of the potential cloud providers when choosing a provider and deploying Teamcenter. Most providers offer services in multiple regions or zones. Cloud features may vary from region to region, or even zone to zone. Choose the region or zone closest to your users that offers the services and features you need.

- Keep in mind that some cloud storage options provide persistent storage, while others are only temporary.

[66] Ingress and egress are respectively network traffic into and out of the cloud facility, usually measured in bytes.

9.2.2 *Sizing Teamcenter in the Cloud*

CPU and memory sizing tends to vary depending on the cloud provider and operating system selected. Refer to the sizing information in the relevant cloud sections that follow.

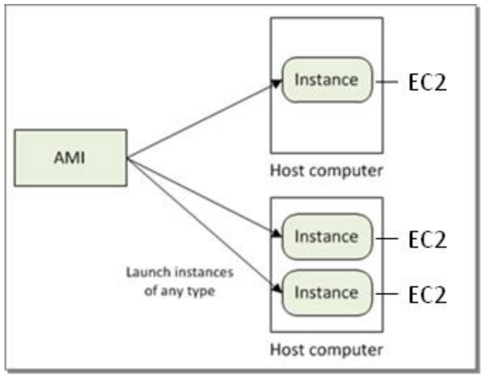
9.3 Teamcenter in Amazon Web Services – AWS

Amazon Web Services provides a variety of basic compute, storage, and network capabilities in its IAAS offering.

9.3.1 *Key AWS Features*

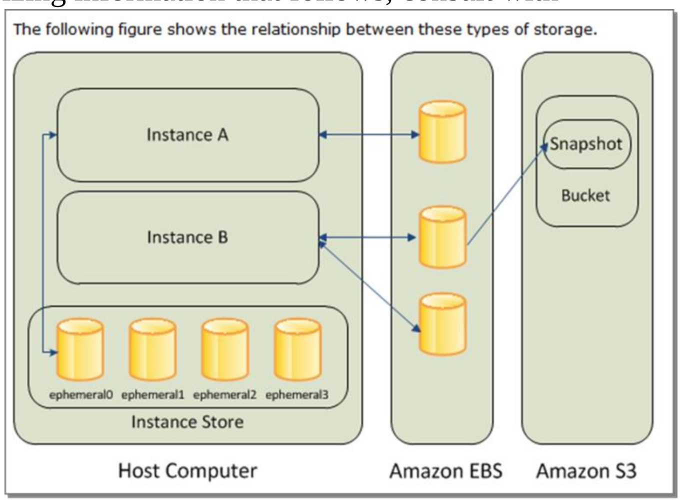
Key AWS features / options for Teamcenter deployment include:

- Amazon Elastic Compute Cloud (EC2): a service that provides re-sizeable computing capacity that you use to build and host your software systems. In AWS, a 'server' is one EC2 compute node.
- Amazon Machine Image (AMI): a template that contains a software configuration (for example, an operating system, an application server, and applications) that can be instantiated in one or more EC2 servers.
- Amazon Storage Options: Instance Store, Elastic Block Storage (EBS), Amazon Simple Storage Service (S3)



Amazon offers a wide range of EC2 compute resources with various amounts of CPU, RAM, and I/O capacity. Together with the sizing information that follows, consult with your AWS account executive to select the most appropriate EC2 option(s).

All EC2 instance types, with the exception of Micro instances, offer instance store, which provides your servers with local, temporary, block-level storage. This is storage that is physically attached to the host computer. The data on an instance store volume doesn't persist when the associated server instance is stopped or terminated. Use instance store only for temporary file systems, e.g. syslogs, database temp tables…



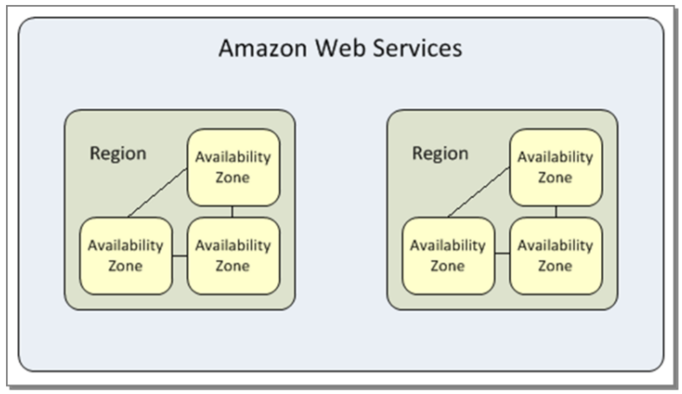
Use Amazon EBS for all Teamcenter persistent storage (e.g. TC_ROOT, TC_DATA, database tablespaces, Tc volumes…). EBS volumes are network-attached, and persist independently from the life of an instance. Amazon EBS volumes can be leveraged as an Amazon EC2 instance's boot partition or attached to a running Amazon EC2 instance as a standard block device. EBS volumes must be crated in the same Availability Zone as the instance to be mountable by the instance. EBS volumes thereafter live and are visible in only that Availability Zone.

Amazon S3 can be used to create point-in-time consistent snapshots of EBS volumes and then automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes, and can protect your data for long term durability. Amazon S3 snapshots are accessible from all the Availability Zones in the region.

9.3.2 *Amazon Zone and Regions*

Amazon EC2 provides the ability to place servers in multiple locations. Amazon EC2 locations are composed of Regions and Availability Zones.

Availability Zones are distinct geographic locations that are engineered to be insulated from failures in other Availability Zones, but provide inexpensive, low latency network connectivity to other Availability Zones in the same Region. By launching instances in separate Availability Zones, you can protect your applications from failure of a single location.



Regions consist of one or more Availability Zones, are more widely geographically dispersed, and will be in separate geographic areas or countries.

The Amazon Elastic Compute Cloud is provided in multiple regions so you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements. Each Amazon region is designed to be completely isolated from the other Amazon regions

Amazon data center facilities are highly available. However, failures can occur that affect the availability of instances that are in the same Availability Zone. Although this is rare, if you host all your Amazon EC2 instances in a single Zone that is affected by such a failure, none of your systems would be available. Design your application across Availability Zones.

9.3.3 *Teamcenter Sizing Guidelines for AWS*

This section provides guidelines for establishing initial server configuration and sizing requirements. It provides information about the types of computing resource required for a specific thin client usage profile[67], and aspects of each that may require adjustments to meet your unique usage requirements.

9.3.4 *Sizing Quick Reference*

This quick reference is to help locate server sizing information quickly for those who have already read this and previous relevant chapters. It assumes the reader has a good understanding of the factors that affect sizing information for the server tiers and terms (e.g. ODR, SDR, Usage Profiles, user categories, etc.) presented in previous chapters.

NOTE

The tables below provide guidelines for initial sizing estimates ***of pilot deployments*** only. Do not use this information until you have reviewed this and other sizing sections in their entirety.

[67] as measured in the Siemens PLM APA Scalability Lab

Table 9-1, AWS Server Sizing Information

Component	**Guideline**
CPU	Assuming the Usage Profile matches APA Thin and Rich Client Benchmarks;
	Peak and average SDR for measured tiers is listed below as the number of AWS Compute Units (ACUs)[68] per user:

	SUSE servers, Thin Client	
Tier	**Peak ACUs / User**	**Avg ACUs / User**
Web	0.0010	0.00028
Enterprise	0.0068	0.00340
Database	0.0023	0.00127
FMS / Flex	0.0001	0.00003

Therefore, multiply the # of users by the average ACU and factor in an Operating Range Reserve of 20% to handle the login rate.

For example, using 1000 users on the database server:

Step 1) **0. 00127** x 1000 = 1.27

Step 2) 1.27 ÷ 80% = 1.5875

Step 3) Select an AWS EC2 instance with 1.6 compute units or more.

Memory — Each *Concurrent* user consumes approximately the amount of RAM listed below in Megabytes. Note that all systems were configured with sufficient RAM to prevent any paging.

	SUSE servers, Thin Client	
Platform	**MB RAM / User**	**MB SWAP / User**
Web	0.680	0
Enterprise	53.30	0
Database	14.89	0
FMS / Flex	0.378	0

.

[68] SPECint_rate values are not available for AWS systems as Amazon does not disclose the actual model information for their installations (like most other cloud providers). Instead, Amazon considers a single 2.0GHz core as a one ‘Amazon Compute Unit’, or ACU. A single 3.0GHz core is considered 1.5 ACUs, etc.

Be sure to review and understand the relevant sections of previous chapters for each tier of a Teamcenter deployment, specifically:

- APA Benchmark Usage Profile
- User Login Rate
- User Types
- User Categories
- Features Used
- Usage Profile Accuracy
- Demand Rate
- Monitor System Usage
- Network Considerations
- Other Consideration

9.3.5 *AWS Server Sizing*

Teamcenter server size varies depending upon the anticipated *Usage Profile*, client type, and server operating system platform.

Four EB2 instance configurations of various sizes were created running Suse 11 SP1 with Teamcenter tiers deployed on each as shown in Figure 9-3. For the purposes of sizing, `lscpu`[69] was used to determine processor information (core count & frequency).

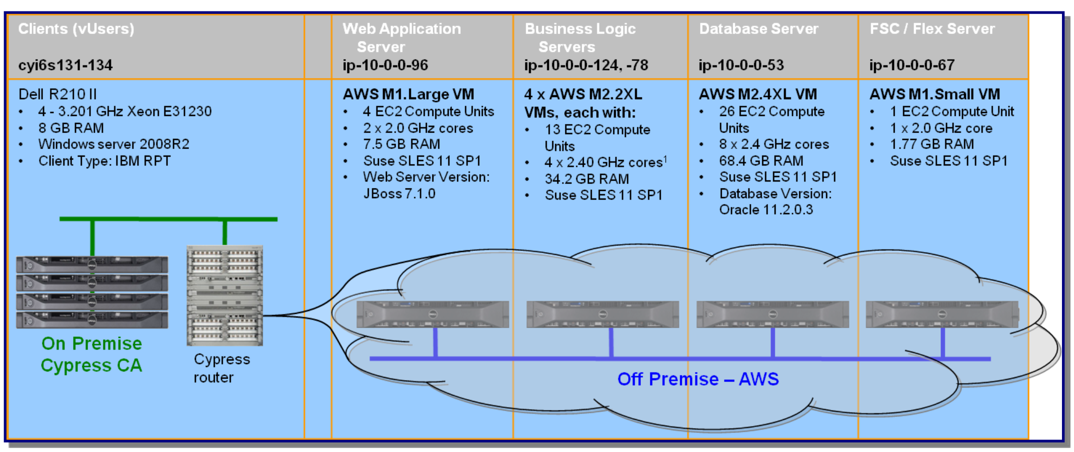


Figure 9-3, AWS Teamcenter Deployment

The APA standard 1000 user Thin Client usage profile was used to derive sizing information for all four Teamcenter tiers in the AWS deployment. Two server pools

[69] lscpu | egrep "socket|Thread”

(Enterprise tier) were deployed to share the 1000 user load. Server utilization was almost identical between the two server pools, therefore only one is depicted in the charts below.

9.3.5.1 AWS Server CPU Sizing

Figure 9-4 illustrates CPU consumption of the EC2 instances for the APA standard Thin Client profile. These charts track CPU consumed as users login, work, and then logout. As more users log in, more CPU is consumed. It is not unusual for the Business Logic server (Enterprise tier) to show additional CPU consumed during the login phase, and the Web server to show additional CPU consumed during the logout phase.

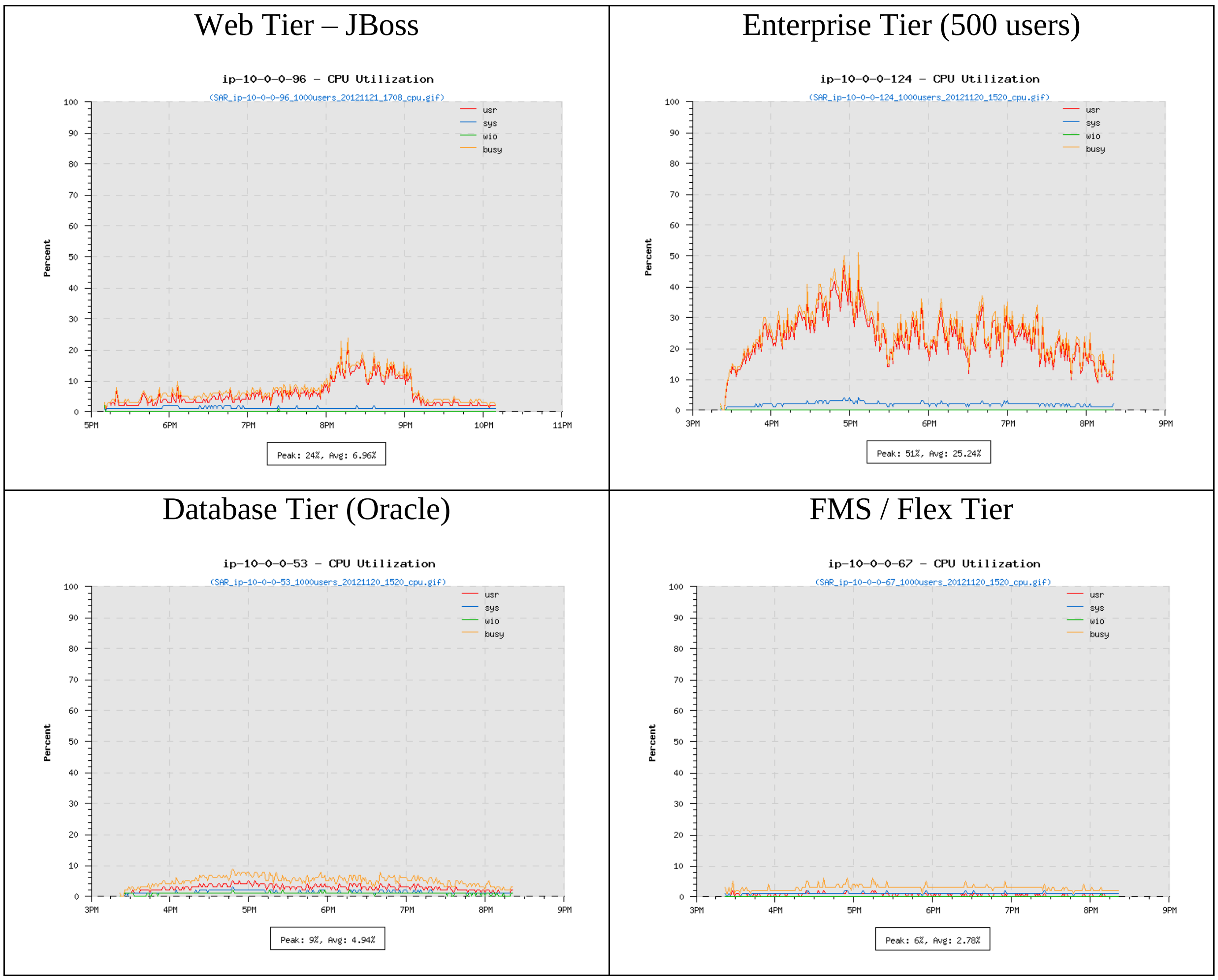


Figure 9-4, AWS Server CPU Utilization – All Tiers

9.3.5.2 AWS Server Memory Sizing

Figure 9-5 illustrates memory consumption of the EC2 instances for the APA standard Thin Client profile. These charts track available memory as users login, work, and then logout (freemem on UNIX). As more users log in, more RAM is consumed. Used memory is calculated as the starting amount before the benchmark starts less the lowest amount observed (Used = Start - Lowest). It is not unusual for the Web and FMS servers to show memory not being released, which would occur after the run at the next Java garbage collection cycle.

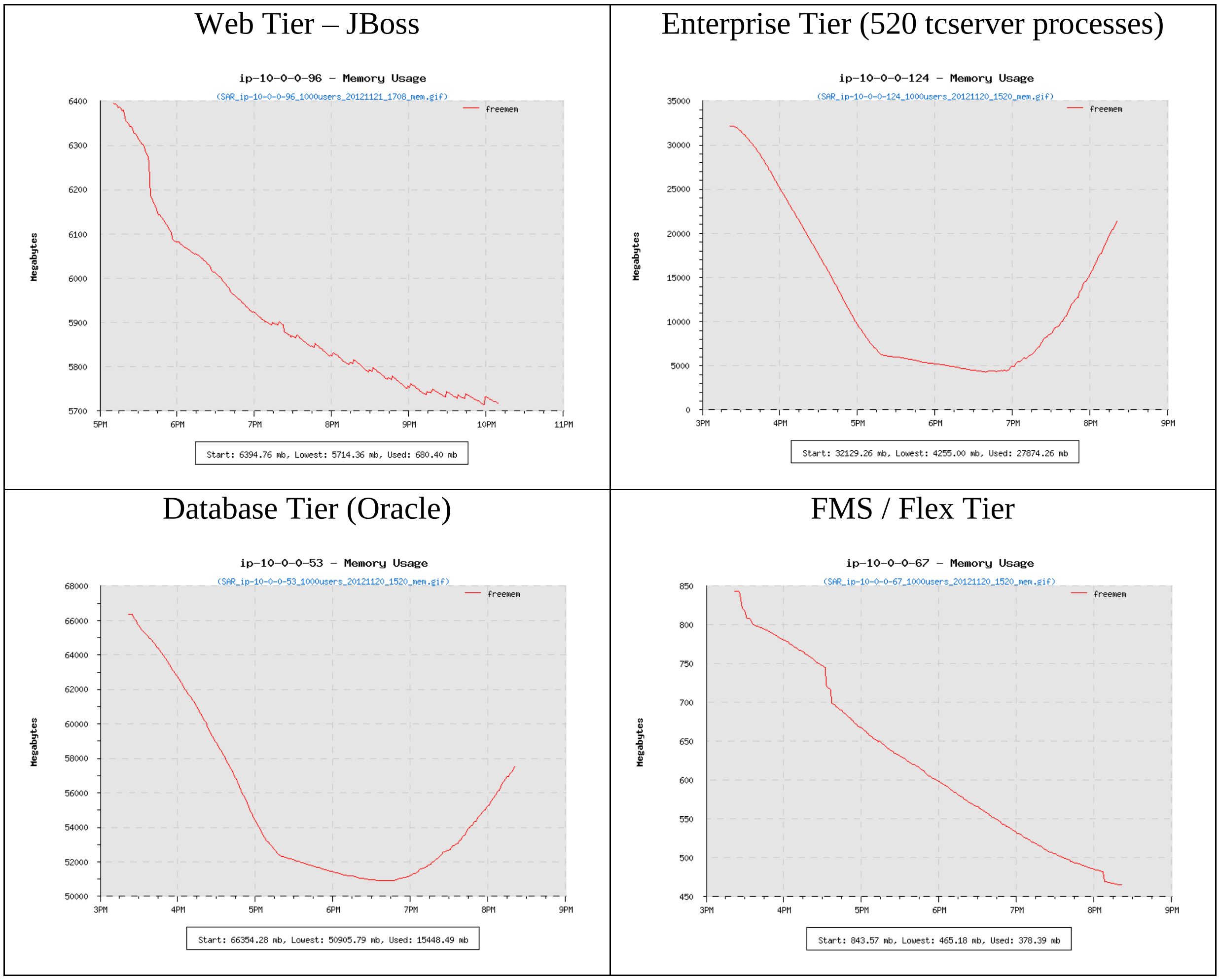


Figure 9-5, AWS Server RAM Utilization – All Tiers

Based on the measurements above, Table 9-2 below provides examples of AWS configuration options, assuming usage profiles similar to the APA standard usage profile.

Table 9-2, Estimated AWS Instance Type Sizes by Number of Users for the Thin Client Profile

<table>
<tr><th>Non-CAD Concurrent Users</th><th>1-50</th><th>50-100</th><th>100-200</th><th>200-500</th><th>500-1000</th><th>> 1000</th></tr>
<tr><td>Web</td><td rowspan="4">M1.Med</td><td rowspan="2">M1.Lrg</td><td>M1.Small</td><td>M1.Small</td><td>M1.Small</td><td rowspan="5">Custom</td></tr>
<tr><td>Server Pool</td><td>M1.XL</td><td>M1.2XL</td><td>2) M2.XL or M2.4XL</td></tr>
<tr><td>Database</td><td rowspan="2">M1.Med</td><td rowspan="2">M1.Lrg</td><td rowspan="2">M2.XL</td><td rowspan="2">M2.4XL + 1000Mb EBS[1]</td></tr>
<tr><td>FMS / Flex</td></tr>
<tr><td>Notes</td><td></td><td>or 1 M2.XL</td><td>or 1 M2.2XL</td><td>or 1 M2.4XK</td><td></td></tr>
</table>

9.3.5.3 AWS Server Swapfile/Pagefile Sizing

To minimize paging impact to server response times, all tiers were configured with sufficient RAM to avoid any paging or swapping. Swapfile / Pagefile sizing is expected to be consistent with on-premise deployments on the same operating system version.

NOTE

To minimize paging and assure optimal performance, limit the number of users per server by the amount of physical memory installed in the machine (see for example section 4.5.3.1 above). The actual swap requirement per user may be different depending upon your production *Usage Profile*.

9.3.5.4 AWS Disk Sizing

Disk storage requirements should be similar to on-premise.

Note that local disk storage (i.e. Instance Storage) is not persistent between instance shutdowns. Use instance storage only for temporary file systems, e.g. syslogs, database temp tables…

Use Amazon Elastic Block Storage (EBS) for all Teamcenter persistent storage (e.g. TC_ROOT, TC_DATA, database tablespaces, Tc volumes…). EBS volumes are network-attached and persist independently from the life of an instance. Amazon EBS volumes can be leveraged as an Amazon EC2 instance's boot partition or attached to a running Amazon EC2 instance as a standard block device.

Disk I/O metrics taken from the Oracle database server with the standard EBS I/O option showed very inconsistent I/O performance over time. Over multiple runs read latencies varied from < 2ms to > 10ms, while write latencies varied from < 1ms to ~ 20ms. Upgrading to Optimized EBS I/O provided consistently better performance than normally observed in the Siemens PLM APA lab[70], with idata tablespace reads consistently under 1ms and write latencies under 50ns.

For most Teamcenter tiers, the standard EBS I/O option will likely prove acceptable. For database servers Siemens PLM Software highly recommends the Optimized EBS I/O option for disk tablespace.

9.3.6 *Impact of Overloading AWS Servers*

Overloading the CPU of any Teamcenter tier, especially the Enterprise tier, will have a direct adverse impact on response times. Be certain to review section 4.5.4, *Impact of Overloading Enterprise Server CPU*, which can be applied to all Teamcenter tiers.

[70] 4 Gbit SAN to high throughput NetApp storage device.

9.4 Teamcenter in Microsoft Azure

Microsoft Azure also provides a variety of basic compute, storage, network and graphics processing capabilities that are used in Infrastructure as a Service (IaaS) deployment offerings. Starting with Teamcenter version 11.2.3.1_a01_2 and later, Teamcenter can be deployed on Microsoft Azure IaaS. Active Workspace 3.3 and later is also supported, however it is recommended that Active Workspace 3.4 is used if deploying the Active Workspace Vis Server Manager for rendering and using IIS as the web application server. Earlier versions of Teamcenter and Active Workspace are not supported for deployment on Microsoft Azure.

9.4.1 *Key Azure Features*

There are a number of key features and services on the Microsoft Azure cloud recommended for use in a Teamcenter IaaS deployment.

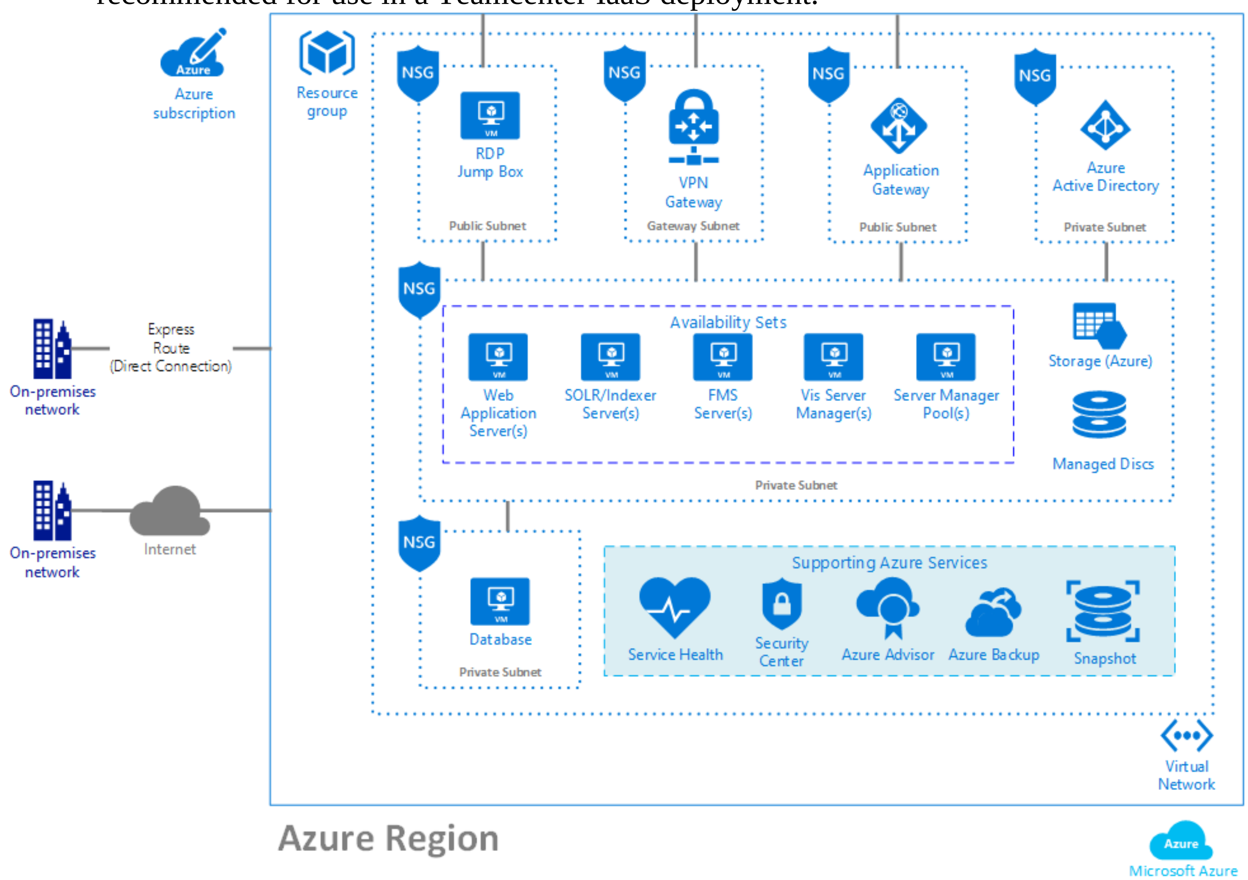


Figure 9-6, Teamcenter on Azure Reference Architecture

Azure Subscription: An Azure Subscription is required in order to consume compute, storage, network and graphics processing capabilities and platform services to support IaaS deployments.

Resource Group: A resource group is a container that holds resources for an Azure solution, or only those resources that need to be managed as a group.

Virtual Network: Azure Virtual Network (VNet) is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription.

Virtual Machines: Azure Virtual Machines (VM) is an on-demand, scalable computing resource that you can use to build and host your software systems. Azure offers a wide range of VM compute resources with various amounts of CPU, RAM, and I/O capacity depending on the workload requirements.

Network Security Groups: A Network Security Group (NSG) contains a list of security rules to filter - allow or deny – network traffic between Azure resources within the Azure Virtual Network. NSGs can be associated with a subnet or a Network Interface Card attached to a Virtual Machines. When a Network Security Group is associated with a subnet, the rules are applied to all resources in the subnet.

Virtual Machine Images: A Virtual Machine Image contains software configuration (ex: an operating system, an application server, and applications) that can be instantiated in one or more Virtual Machines.

Storage Options: Azure Storage is a Microsoft-managed cloud service that provides storage that is highly available, secure, durable, scalable, and redundant. The storage options commonly used are Azure File Storage and Managed Disks. All VM instance type servers are provided with local temporary storage. This is storage that is physically attached to the host computer.

Application Gateway: A dedicated virtual appliance providing application delivery controller (ADC) as a service, offering various layer 7 load balancing capabilities for your application and a Web Application Firewall (WAF).

Express Route: ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection facilitated by a connectivity provider.

Azure Active Directory: Azure Active Directory is Azure's multi-tenant, cloud based directory and identity management service. Azure AD combines core directory services, advanced identity governance, and application access management.

Virtual Private Network (VPN) Gateway: Azure VPN Gateway is a type of virtual network gateway appliance that sends encrypted traffic between Azure Virtual Network and an on-premises datacenter. The encrypted traffic is sent over a public connection.

Accelerated Networking: By default the traffic coming in and out of a Virtual Machine traverses through the host and a virtual switch. With Accelerated Networking the traffic arrives directly at the virtual machine NIC and the Virtual Machine rather through a virtual switch resulting in lower latency, reduced jitter and decreased CPU utilization. Accelerated Networking must be enabled during Virtual Machine creation. It cannot be enabled after Virtual Machine creation.

There are also a number of Azure services that are used to support IaaS deployments:

Service Health: Provides timely and personalized information when problems or events in Azure services impact your services and require attention and helps in preparing for upcoming planned maintenance on Azure.

Azure Security Center: This is a security management service that provides advanced threat protection. With Azure Security Center, you can continuously monitor the security of your machines, networks and Azure services, limit exposure to threats, and detect and respond swiftly to attacks.

Azure Advisor: Azure Advisor is a personalized recommendation engine that provides proactive best practices guidance for optimally configuring your Azure resources.

Azure Backup: Backup and Restore Azure Virtual Machines and individual files or folders as needed. Azure Backup can provide both short and long-term retention of your data.

Azure offers various Virtual Machines types, families and sizes to choose from depending on the application workload. Together with the sizing information that follows, consult with your Azure account executive to select the most appropriate Virtual Machine options.

Each Azure VM contains an operating systems disk and temporary disk. The temporary disk labeled as D: drive by default on Windows VMs provides a short-term storage that may be lost during a maintenance event or when the VM is redeployed. The data temporary drive is persisted during a standard reboot.

Azure Managed Disks can be used for all persistent storage (e.g. TC_ROOT, TC_DATA, database tablespaces, Tc volumes). Further information about disk storage for Azure Virtual Machines see here: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/about-disks-and-vhds. This link contains information covering managed disks, premium storage, standard storage as well as performance and scalability targets.

With managed disks, only the disk size and type (Standard or Premium) are specified and Azure will create the managed disk. This is in contrast to unmanaged disks which has limitations on the number of disks and needs to be always linked with a storage account.

Managed disks do not have size limits or storage account limits. The disk is independent of the virtual machine itself and it is very easy to take a snapshot (i.e. copy) of the disk. A new virtual machine can then also be created using the snapshot. A managed disk can be created as an empty disk or based on a Blob or snapshot and then attached to a virtual machine.

For hosting Teamcenter volumes, it is recommended to attach multiple premium disks to a Virtual Machine and stripe them together to get a combined higher Input/output operations per second (IOPS) and throughput limit. On a DS series Virtual Machine a maximum of 32 premium disks can be striped together and for a GS series up to 64 premium disks can be striped. The only limiting factor here is that the combined IOPS cannot be more than the limit on the IOPS of the Virtual Machine SKU. On Windows, another option is that the disks can be striped together using Storage Spaces Direct. Azure also provides a quick start template to create a Storage Spaces Direct (S2D) Scale-Out File Server (SOFS) cluster using Windows Server 2016 in an existing VNET and Active Directory environment.

Azure File Storage contains four types of storage – Blobs, Files, Tables, and Queues. File and Blob storage are commonly used types for Teamcenter deployment. While Blob storage stores unstructured data, File storage provides shared storage that can be mounted as shared paths using the SMB protocol on Windows or Linux, and allows creating a hierarchical folder structure to upload files. The Azure File Storage has certain advantages such as high availability, encryption and connectivity. But, File Storage is not recommended to use for storing Teamcenter Volumes due to the IOPS limitation. Each file share is limited to only 1000 IOPS. This file storage option can be used for storing Teamcenter kits and other such software that need to be downloaded on multiple virtual machines.

9.4.2 ***Azure Regions and Availability***

Azure operates in multiple datacenters around the world. Azure datacenters grouped into multiple geographic regions give users the flexibility in choosing where to deploy the applications. Azure is generally available in 32 regions (and counting) around the world to enable the deployment of applications closest to the users and satisfy the requirements regarding the location of data. The multiple datacenters that exist within each region provide redundancy and availability. Azure also has special regions specific to Gov Cloud satisfying special compliance and legal requirements. The features and services available could differ between various regions.

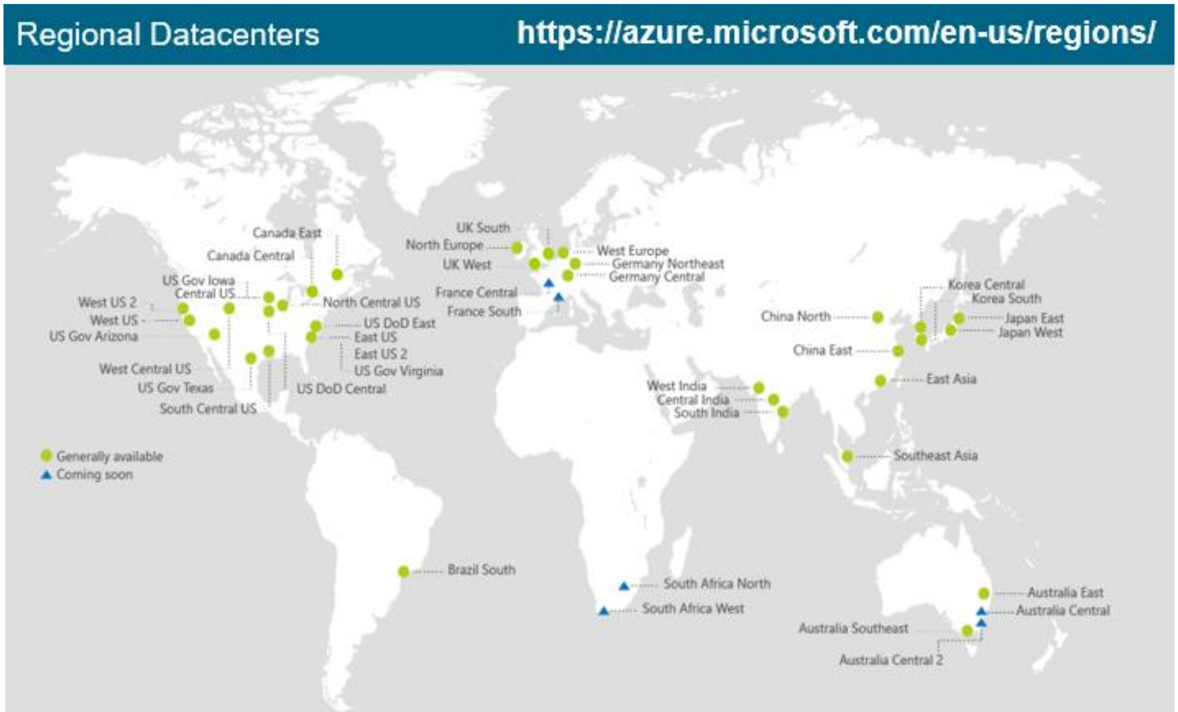


Figure 9-7, Microsoft Azure Regions

An Azure availability set is a logical grouping of VMs within a datacenter that allows Azure to understand how applications are built to provide for redundancy and availability. Azure recommends that two or more VMs are created within an availability set to provide for a highly available application and to meet the 99.95% Azure SLA. VMs deployed within an availability set, are automatically distributed across different fault and update domains. A fault domain is a logical group of underlying hardware that share a common power source and network switch, similar to a rack within an on-premises datacenter. An update domain is a logical group of underlying hardware that can undergo maintenance or be rebooted at the same time. VMs deployed in an availability set are thus protected against power and network failures occurring to racks in datacenters and do not undergo maintenance updates at the same time. There is also a single instance VM SLA of 99.9% available when using premium storage for both operating system and data disks, which can help reduce the need for using Availability Sets.

Availability zones (in preview at the time of this writing) is a feature of Azure that is an alternative to availability sets. Availability zones expand the level of control in maintaining availability of the applications and data on your VMs. Each Azure region has three Availability Zones. Each Availability Zone is logically separated from the other Availability Zones within the same region. They have a distinct power source, network, and cooling. If Teamcenter is deployed with replicated VMs in multiple Availability Zones, then the loss of a datacenter would not impact the availability of the application and data.

9.4.3 *Teamcenter Sizing Guidelines for Azure*

This section provides guidelines for establishing initial server configuration and sizing requirements. It provides information about the types of computing resource required for a specific Active Workspace usage profile[71], and aspects of each that may require adjustments to meet your unique usage requirements.

Azure provides a range of types, families and sizes of virtual machines to choose from to support your compute requirements. Currently, Microsoft offers six virtual machine types to choose from based on the type of workload, published at https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes and summarized below:

- **General Purpose** – Balanced CPU-to-memory ratio
- **Compute Optimized** – High CPU-to-memory ratio
- **Memory Optimized** – High memory-to-CPU ratio
- **Storage Optimized** – High disk throughput and IO
- **GPU** – Specialized virtual machines for heavy graphics rendering and video editing
- **High Performance Compute** – Fastest, most powerful CPU with optional high-throughput network interfaces (RDMA)

Each virtual machine type is available in a number of different families and sizes, identified respectively by a letter and number. Examples include: F3, F4s, DS_v2, H8m or NV24. See the following Microsoft documentation for the full comprehensive list of virtual machine types available for both Windows and Linux platforms:

a) https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes
b) https://docs.microsoft.com/en-us/azure/virtual-machines/linux/sizes

The following table identifies the Azure virtual machine families and types that are generally recommended for Teamcenter tiers.

Table 9-3, Azure VM Sizing Families recommended for Teamcenter Tiers

Series	VM Size Family	Teamcenter deployment use
F, FS Series	**Compute Optimized Virtual Machines** providing Faster CPUs with lower memory or temporary storage	File Management Server (FMS) FlexLM License Server Web Servers Dispatcher (Client, Scheduler, Modules)
DS_v2 Series	**Next Generation Memory Optimized Virtual Machines** providing faster CPUs, better temporary storage and higher memory demands	Database Server Active Workspace SOLR Indexer

71 as measured in the Siemens PLM APA Scalability Lab

Table 9-3, Azure VM Sizing Families recommended for Teamcenter Tiers (cont.)

Series	VM Size Family	Teamcenter deployment use
H Series	**High Performance Compute Virtual Machines** providing more CPU to memory ratio for compute and network intensive applications.	Server Pool Manager
NV Series	**GPU Optimized Virtual Machines** providing GRID and Nvidia Tesla M60, M80 GPU enabled instances	Active Workspace Visualization Server Manager

Microsoft have created the concept of the Azure Compute Unit (ACU) to provide a way of comparing compute (CPU) performance across Azure SKUs. This can help in identifying which SKU is most likely to satisfy your performance needs. ACU is currently standardized on a Small (Standard_A1) virtual machine being 100 and all other SKUs then represent approximately how much faster that SKU can run a standard benchmark. For further information on use of ACU see https://docs.microsoft.com/en-us/azure/virtual-machines/windows/acu.

For example, the DS_v2 series SKU CPU performance has a ACU score of 210-250 means they are up to 2.5 times more powerful than the baseline Standard_A1 SKU with a score of 100. If ACU data is not published for Azure server families or SKU it is recommended the reader contact Microsoft to obtain these values.

Table 9-4, Azure CPU Performance ACU scores[72]

SKU Family	ACU/Core
ExtraSmall	50
Small-ExtraLarge	100
A5-7	100
A8-A11	225*
A v2	100
D	160
D v2	160 - 190*
D v3	160 - 190*
E v3	160 - 190*
G	180 - 240*
H	290 - 300*

[72] published by Microsoft

ACUs marked with a * use Intel® Turbo technology to increase CPU frequency and provide a performance boost. The amount of the boost can vary based on the VM size, workload, and other workloads running on the same host.

The calculated ACU score published by Microsoft is dependent upon the physical CPUs used on the server families. For example, the DS_v2 series Virtual Machines employ the Intel® Xeon® E5-2673 v3 (Haswell) processor. The published ACU score for this machine family is 210 – 250. A minimum of 210 ACU is guaranteed but use of Intel's Turbo technology can allow a peak of 250 ACU. When comparing to on-premises hardware with the same CPU and assuming no overprovisioning, these can be compared 1:1.

When sizing Virtual Machine server configurations and SKUs to use on Azure cloud, there are a number of characteristics that must be taken into consideration:

1) CPU to Memory ratio is fixed on available virtual machine SKUs
2) Some virtual machine SKUs, especially lower specifications, have limitations as to the number and types of disks they can use as the number of disks depends on the size of the virtual machine or may not have premium (SSD-based) storage options.
3) The virtual CPU to physical CPU core ratio (vCPU:Core) is typically 1:1 for unless hyper-threading is in use, therefore Virtual Machine CPUs are typically not overprovisioned – 1 physical core supports a single virtual core.
4) Some Virtual Machine type series, including F, Fs and H, utilize Intel® Turbo technology to increase CPU frequency and provide a performance boost.
5) The number of Network Interface Cards (NICs) and bandwidth. All Virtual Machines have can have at least 2 NICs.
6) Consider caps that are placed on Virtual Machine family SKUs by Azure around memory, local storage, number and size of data disks along with their associated throughput as well as NICs.

Virtual Machine sizes on Azure are published by Microsoft here:
https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-sizes-specs

The following example table is taken from the above published sizing page and shows the sizing information for the D_V2 family of Virtual Machine SKUs. This shows the types of configurations available and the caps placed on the Virtual Machines.

Table 9-5, Azure D_V2 Server Family SKU sizes published by Microsoft

Size	**CPU cores**	**Memory: GiB**	**Local SSD: GiB**	**Max NICs / Network bandwidth**
Standard_D1_v2	1	3.5	50	1 / moderate
Standard_D2_v2	2	7	100	2 / high
Standard_D3_v2	4	14	200	4 / high
Standard_D4_v2	8	28	400	8 / high
Standard_D5_v2	16	56	800	8 / extremely high
Standard_D11_v2	2	14	100	2 / high

Table 9-5, Azure D_V2 Server Family SKU sizes published by Microsoft (cont.)

Size	CPU cores	Memory: GiB	Local SSD: GiB	Max NICs / Network bandwidth
Standard_D12_v2	4	28	200	4 / high
Standard_D13_v2	8	56	400	8 / high
Standard_D14_v2	16	112	800	8 / extremely high
Standard_D15_v2	20	140	1,000	8 / extremely high

9.4.4 *Sizing Quick Reference*

This quick reference is to help locate server sizing information quickly for those who have already read this and previous relevant chapters. It assumes the reader has a good understanding of the factors that affect sizing information for the server tiers and terms (e.g. ODR, SDR, Usage Profiles, user categories, etc.) presented in previous chapters.

Microsoft publishes SPECint®_rate2006 (SIR) benchmarks for current server families and SKUs available on Azure at https://docs.microsoft.com/en-us/azure/virtual-machines/windows/compute-benchmark-scores. If benchmark data is not published for Azure server families and SKUs being reviewed it is recommended the reader contact Microsoft to obtain these values. In the future Microsoft may decide to not publish SPEC values at all, at which point users will need to rely on the Azure Compute Unit (ACU) mentioned above.

The majority of the server families and SKUs available on Azure cloud are also used for supported Linux operating system-based Virtual Machines.

NOTE

SPECint_rate values are not available for all Azure systems. If the processor contained therein is not listed in the compute benchmark scores site listed above, nor the SPEC CPU2006 benchmark published results, nor Intel's Automated Relational Knowledge Base (ARK - ark.intel.com), it is not possible to estimate SIR CPU requirements. Normally if a SPEC benchmark is not available for the exact configuration measured, then one is chosen for a system with similar processor characteristics from the ARK specifications, then adjusted for differences (e.g. enabled cores, speed…). But without an entry in the ARK system it is not possible to do this.

In these cases the percentage of CPU usage is listed rather than a SIR guideline.

NOTE

The tables below provide guidelines for initial sizing estimates ***of pilot deployments*** only. Do not use this information until you have reviewed this and other sizing sections in their entirety. Note that the Active Workspace usage profile is considerably different from the thin client usage profile used in the previous cloud section.

Table 9-6, Azure Server Sizing Information

Component **Guideline**

CPU Assuming the Usage Profile matches APA Active Workspace Client Benchmarks;

Peak and average SDR for measured tiers is listed below where available as SPECint®_rate2006 (SIR) per user. Where SIR was not available the percentage of total CPU is given:

			Windows servers, Active Workspace Client		
Tier	**Azure SKU**	**VMs**	**Peak SIR / User**	**Avg SIR / User**	**Processor** (see Figure 9-8)
Web	F4	1	9.78%	3.30%	2 cores E5-2673 v3 @ 2.40GHz
Enterprise	H16	3	0.0383	0.0119	8 cores E5-2667 v3 @ 3.20 GHz
Visualization	NV24	2	0.0610	0.0191	12 cores E5-2690 v3 @ 2.60 GHz
Database	DS5_V2	1	13.11%	6.19%	8 cores E5-2673 v3 @ 2.40GHz
Solr	F16s	1	9.73%	3.81%	8 cores E5-2673 v3 @ 2.40GHz
FMS / Flex	F16s	1 ea.	0.66%	0.21%	8 cores E5-2673 v3 @ 2.40GHz

Table 9-6, Azure Server Sizing Information (cont.)

Component	Guideline
CPU	**NOTE** Where multiple systems are pooled together (i.e. Enterprise pool and Visualization pool) users are distributed fairly evenly across the pooled VMs. The peak values given are the highest peak values of all VMs in that pool while the average values given are average for the pool. If you plan to configure a single VM for all tiers adjust these values accordingly. Therefore, where SIR are available, multiply the # of users by the average SIR and factor in an Operating Range Reserve of 20% to handle the login rate. For example, using 1000 users on the visualization server: Step 1) **0. 061** x 1000 = 6.1 Step 2) 6.1 ÷ 80% = 7.625 Step 3) Select an Azure instance with a SIR of 7.625 or more. Where SIR are not available make a similar estimate using CPU percentage with a similar speed and architecture processor.
Memory	Each *Concurrent* user consumes approximately the amount of RAM listed below in Megabytes. Note that all systems were configured with sufficient RAM to prevent any paging.

	Windows servers, Active Workspace Client	
Platform	**MB RAM / User**	**MB SWAP / User**
Web	0.53	0
Enterprise	149.43	0
Visualization	19.10	0
Database	0.02	0
Solr	0.69	0
FMS / Flex	0.439	0

NOTE

Where multiple systems are pooled together (i.e. Enterprise pool and Visualization pool) users are distributed fairly evenly across the pooled VMs. RAM used is calculated as the total used in all pool machines divided by the total number of users.

Be sure to review and understand the relevant sections of previous chapters for each tier of a Teamcenter deployment, specifically:

- APA Benchmark Usage Profile
- User Login Rate
- User Types
- User Categories
- Features Used
- Usage Profile Accuracy
- Demand Rate
- Monitor System Usage
- Network Considerations
- Other Consideration

9.4.5 *Azure Server Sizing*

Teamcenter server size varies depending upon the anticipated *Usage Profile*, client type, and server operating system platform.

Azure configurations of various sizes were created running Windows 2012R2 Datacenter edition with Teamcenter tiers deployed on each as shown in Figure 9-3. For the purposes of sizing, Windows `logman` was used to determine processor information (core count & frequency).

Web Application Server	Business Logic Servers (3)	Vis Servers (2)	Database Server (SQL Server)	Solr Index Server	FMS Server	Flex Server
Standard F4	**Standard H16**	**Standard NV24**	**Standard DS5 V2**	**Standard DS5 V2**	**Standard F16s**	**Standard F16s**
WebServer0 • Microsoft Corporation Virtual Machine • 4 vCPUs (Intel® Xeon® CPU E5-2673 v3 @ 2.40GHz) • 8 GB RAM • Microsoft Windows Server 2012 R2 Datacenter • Microsoft .NET Framework 4.0.30319 • Microsoft Internet Information Services Version 8.5 • http://webserver0-g:7001/tc/webclient	**ServerPoolMgr0-2** • Microsoft Corporation Virtual Machine • 16 vCPUs (Intel® Xeon® CPU E5-2667 v3 @ 3.20GHz) • 112 GB RAM • Microsoft Windows Server 2012 R2 Datacenter • Microsoft .NET Framework 4.0.30319	**VisServer0-1** • Azure Standard NV24 • 24 vCPUs (E5-2690 v3) • 224GB RAM • Win2012R2 DataCenter • 4 x GPU (2 x NVIDIA M60 cards)	**DBServer** • Microsoft Corporation Virtual Machine • 16 vCPUs (Intel® Xeon® CPU E5-2673 v3 @ 2.40GHz) • 56 GB RAM • Microsoft Windows Server 2012 R2 Datacenter • Microsoft SQL Server 2012 • M12TCwr	**Indexer0** • Microsoft Corporation Virtual Machine • 16 vCPUs (Intel® Xeon® CPU E5-2673 v3 @ 2.40GHz) • 56 GB RAM • Microsoft Windows Server 2012 R2 Datacenter	**fmsmaster0** • Microsoft Corporation Virtual Machine • 16 vCPUs (Intel® Xeon® CPU E5-2673 v3 @ 2.40GHz) • 32 GB RAM • Microsoft Windows Server 2012 R2 Datacenter	**LicenseServer0** • Microsoft Corporation Virtual Machine • 16 vCPUs (Intel® Xeon® CPU E5-2673 v3 @ 2.40GHz) • 32 GB RAM • Microsoft Windows Server 2012 R2 Datacenter

Figure 9-8, Azure Teamcenter Deployment

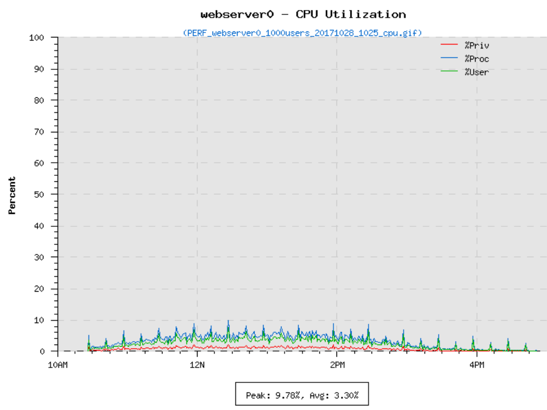
The APA standard 1000 user Active Workspace Client usage profile was used to derive sizing information for all Teamcenter tiers in the Azure deployment. Three server pools (Enterprise tier) and two vis server pools (Visualization tier) were deployed to share the

1000 user load. Server utilization was almost identical on each of the VMs in the server pool, therefore only one is depicted in the charts below.

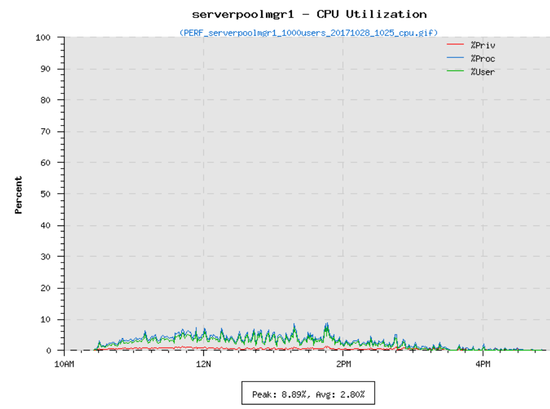
9.4.5.1 AZURE Server CPU Sizing

Figure 9-4 illustrates CPU consumption of the various VMs for the APA standard Active Workspace Client profile. These charts track CPU consumed as users login, work, and then logout. Where multiple VMs were deployed for a single tier, the chart shown is for the highest peak CPU of the set. As more users log in, more CPU is consumed.

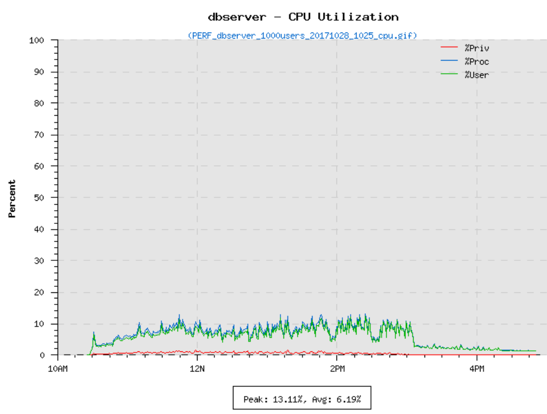
Web Tier (IIS/.net)



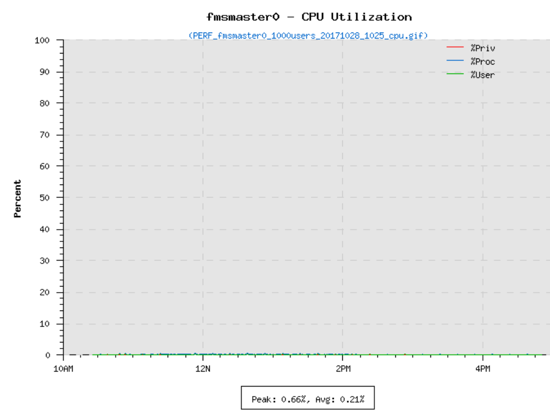
Enterprise Tier (1 of 3)



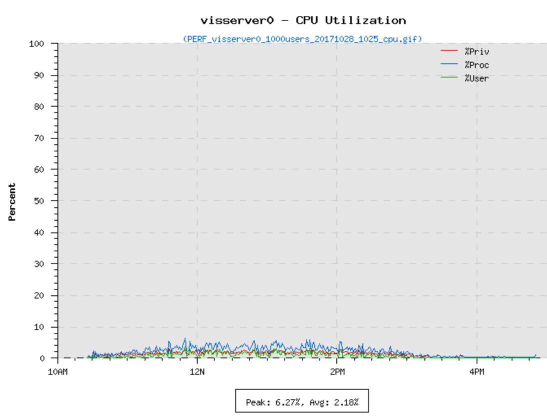
Database Tier (SQL Server)



FMS[73]



Visualization (1 of 2)



Solr

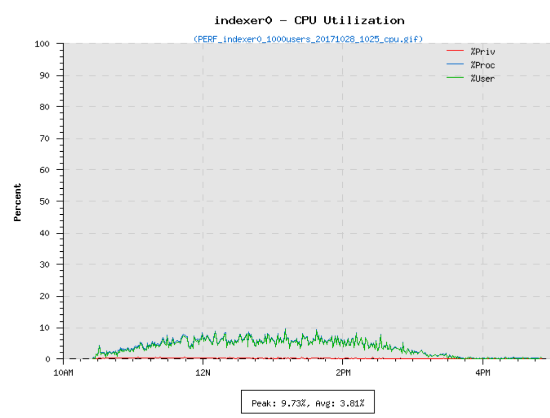


Figure 9-9, Azure Server CPU Utilization – All Tiers

9.4.5.2 Azure Server Memory Sizing

Figure 9-5 illustrates memory consumption of the Azure VMs for the APA standard Active Workspace Client profile. These charts track available memory as users login, work, and then logout. As more users log in, more RAM is consumed. Used memory is calculated as the starting amount before the benchmark starts less the lowest amount observed (Used = Start - Lowest).

[73] FlexLM license server on a separate VM and used virtually no CPU.

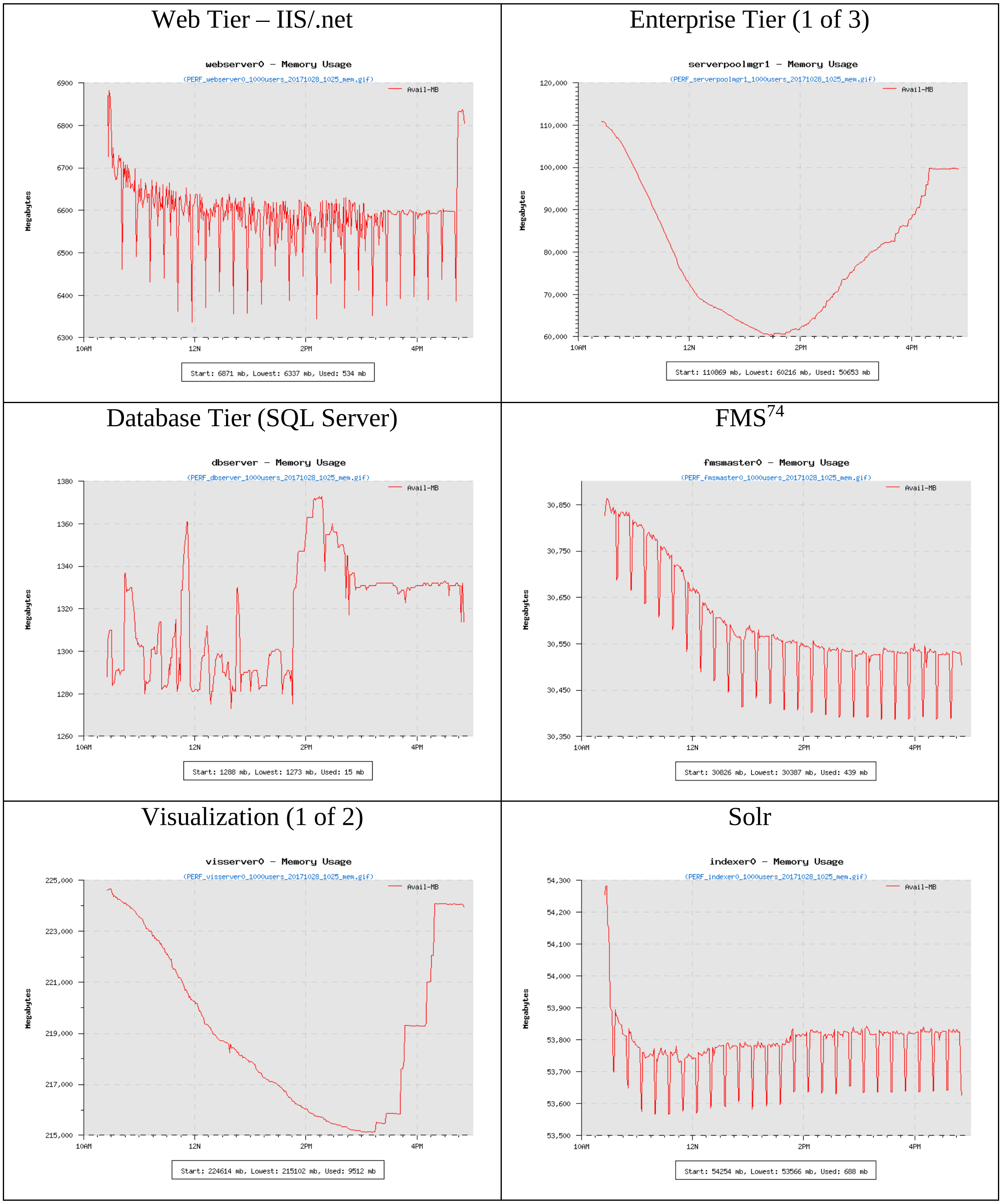


Figure 9-10, Azure Server RAM Utilization – All Tiers

[74] FlexLM license server on a separate VM and used virtually no CPU.

Based on the measurements above, Table 9-2 above provides examples of Azure configuration options, assuming usage profiles similar to the APA standard usage profile.

9.4.5.3 Azure Server Pagefile Sizing

To minimize paging impact to server response times, all tiers were configured with sufficient RAM to avoid any paging. Pagefile sizing is expected to be consistent with on-premise deployments on the same operating system version.

NOTE

To minimize paging and assure optimal performance, limit the number of users per server by the amount of physical memory installed in the machine (see for example section 4.5.3.1 above). The actual swap requirement per user may be different depending upon your production *Usage Profile*.

9.4.5.4 Azure Disk Sizing

Disk storage requirements should be similar to on-premise.

The Standard managed disks on Azure used by Virtual Machines use HDD as storage media. Microsoft Azure recommends using Standard disks for development, test, and other infrequent access workloads that are not sensitive to performance variability. The standard disks provide 500 IOPS and a throughput of 60MB/sec per disk.

For production workloads, Microsoft recommends using Premium Managed Disks. Premium disks use high performance Solid State Drive (SSD) as storage media. They are designed to support I/O intensive workloads with significantly high throughput and low latency. The IOPS and throughput of a premium disk varies per size. For example, a P10 disk of size 128GB provides 500 IOPS and 100 MB/sec throughput whereas a P20 disk of size 512GB provides 2300 IOPS and 150MB/sec throughput. Premium Managed Disks are supported by DS-series, DSv2-series, FS-series, and GS-series VM sizes which are specifically targeted for Premium Disks.

It is therefore recommended that when determining disk storage requirements on Azure for Teamcenter Volumes, the IOPS performance required is considered along with the total storage space requirements.

The following table outlines Standard disk limits (managed and unmanaged) in Azure:

Table 9-7, Azure Standard Storage disks limits

VM Tier	Basic Tier VM	Standard Tier VM
Max Disk size	4095 GB	4095 GB
Max 8 KB IOPS per disk	Up to 300	Up to 500
Max Bandwidth per disk	Up to 60 MB/s	Up to 60 MB/s

The following table outlines Premium disks limits in Azure:

Table 9-8, Azure Premium Storage disks limits

Premium Disks Type	**P4**	**P6**	**P10**	**P20**	**P30**	**P40**	**P50**
Disk size	32 GB	64 GB	128 GB	512 GB	1024 GB	2048 GB	4095 GB
IOPS per disk	120	240	500	2300	5000	7500	7500
Throughput per disk (per second)	25 MB	50 MB	100 MB	150 MB	200 MB	250 MB	250 MB

Example: a DS5_v2 can have a maximum of 64 data disks, whereas a DS1_v2 can have a maximum of four, as shown in the table below.   Therefore, a DS1_v2 can add four P50 (7500 IOPS and 250MB/s). This would render a total 30,000 IOPS and 1GB/s throughput.   However, it should be noted that there are caps on the Virtual Machine level too, which means looking at the uncached statistics the IOPS limit 3,200 and 48MB/s throughput. Therefore the DS1_v2 is not capable of leveraging the capabilities of four P50 disks. The key consideration here is to pay close attention to the limits specified for the Virtual Machine SKUs being considered when sizing storage and storage throughput performance.

Table 9-9, Azure DS_v2 Virtual Machine SKU sizing data published by Microsoft

ACU: 210-250

Size	vCPU	Memory: GiB	Temp storage (SSD) GiB	Max data disks	Max cached and temp storage throughput: IOPS / MBps (cache size in GiB)	Max uncached disk throughput: IOPS / MBps	Max NICs / Expected network bandwidth (Mbps)
Standard_DS1_v2	1	3.5	7	4	4,000 / 32 (43)	3,200 / 48	2 / 750
Standard_DS2_v2	2	7	14	8	8,000 / 64 (86)	6,400 / 96	2 / 1500
Standard_DS3_v2	4	14	28	16	16,000 / 128 (172)	12,800 / 192	4 / 3000
Standard_DS4_v2	8	28	56	32	32,000 / 256 (344)	25,600 / 384	8 / 6000
Standard_DS5_v2	16	56	112	64	64,000 / 512 (688)	51,200 / 768	8 / 12000

9.4.6 *Impact of Overloading Azure Servers*

Overloading the CPU of any Teamcenter tier, especially the Enterprise tier, will have a direct adverse impact on response times. Be certain to review section 4.5.4, *Impact of Overloading Enterprise Server CPU*, which can be applied to all Teamcenter tiers.

Also note that if you use a ‘jumpbox’ VM to access / manage your environment be sure to configure that with adequate resources as well. An overloaded jumpbox can cause the rest of the environment to appear unreliable.

Chapter

10 Improving Teamcenter Performance

Go to main Table of Contents

Chapter

10 Improving Teamcenter Performance

This chapter is intended to help you:

- Understand more about performance issues and configuration decisions that can impact performance.
- Track performance and system statistics to monitor trends that may forewarn of potential problems.
- Isolate performance issues and bottlenecks to specific components of the Teamcenter system and correct them.
- Establish a process for controlling and tracking changes to the Teamcenter environment.
- Maintain optimal response time performance.

This chapter has not been updated from the previous Teamcenter release.

NOTE

Information to improve performance of specific Teamcenter components, including quick reference configuration and performance tuning checklists, is contained the chapter for that component. See the *Table of Contents*.

This chapter is organized as follows:

Section 10.1	General Performance and Scalability information, monitoring and analyzing performance issues.
Section 10.2	A discussion of the various components of a Teamcenter installation, both software and hardware, and configuration options that can contribute to performance or scalability issues.

NOTE

The information in this chapter has not changed for this revision of Deployment Guide. Therefore the information in this section is taken from the Tc 8.3 revision, also available at: *http://support.industrysoftware.automation.siemens.com/docs/teamcenter/8.3/TcDeploymentGuide_Tc8.3j.pdf*

10.1 General Performance Issues

The performance of systems like Teamcenter is affected by several factors, mainly by the specific applications being used and the usage patterns within your business. In general, performance is measured by the response time the end user perceives. Performance issues can be catastrophic with the entire system virtually unusable, or they may be subtle, with only slight response delays occurring in a single application. In many cases, performance symptoms are persistent, although some are observed only occasionally, or only at certain times of the day.

Siemens PLM Software maintains a Performance/Scalability Lab and endeavors to identify response time and scalability issues prior to release. However, issues occasionally are not found before the release is distributed. Performance issues in released Teamcenter software are corrected in subsequent releases as they are found or in the next Maintenance Pack (MP) or Service Release (SR) whenever possible. Consult the *Global Technical Access Center* (GTAC) MP/SR download site, the UGSolutions Web site, or scan Software Field Bulletins (SFBs) for performance related fixes as they become available.

Although all Teamcenter component tiers can be deployed on the same machine, in general performance can be optimized if each component was deployed on a separate server. This allows each server to be tuned for that specific component. A notable exception is that Web Application server and Enterprise Pool Server can be deployed on the same machine.

10.1.1 *Response Time vs. Load vs. Capacity*

Response time is determined by the load generated by users and the system resources available to handle that load. Conceptually, response time is determined by the following formula:

Response time = Resources ÷ Load

Response time is not directly proportional to either resources or load. However, as the load increases, response times may increase. In fact, response times can increase in a very nonlinear manner as system capacity of one or more components approaches 90-95 percent. Adding resources (more or faster CPUs, memory, and so forth) can usually decrease response times (Figure 10-1). In general, response time of Teamcenter remains fairly consistent when system capacity is below 70-80%.

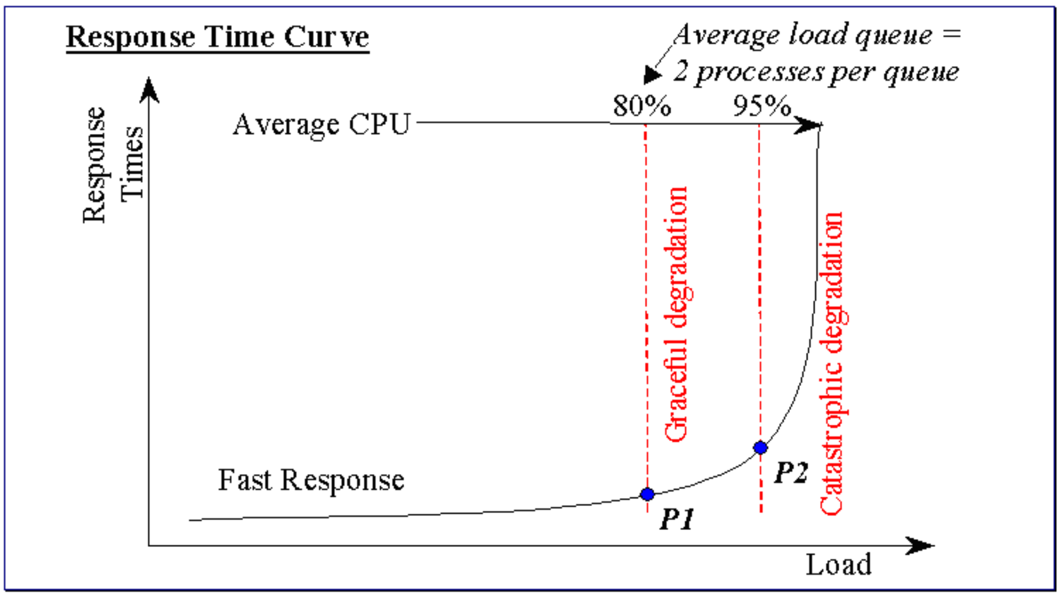


Figure 10-1, Load vs. Response Time

Similarly, the system's ability to move data to and from the users is affected by load. As the system’s input/output (I/O) resource usage exceeds 80–90 percent, data transfer rates begin to decline (Figure 10-2). Again, keeping the system's data transfer rates below 80% of capacity helps maintain consistent data throughput.

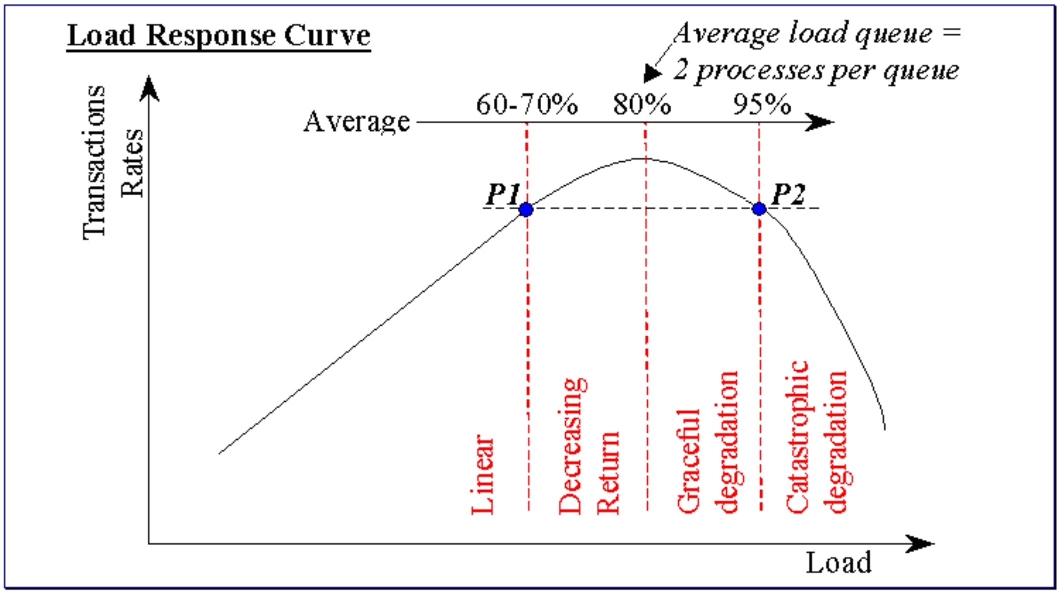


Figure 10-2, Load vs. Throughput

10.1.2 *Performance Symptoms*

Teamcenter symptoms can vary widely because it is a complex, distributed, client-server system. Teamcenter is implemented with several autonomous, yet interdependent components, each of which may contain multiple interdependent processes. This establishes a complicated system of queues and response curves that interact and can behave unpredictably, especially under heavy system load. Performance issues with any single component of the system can have a significant impact on the response time observed by the end user.

When performance issues manifest in more than one component or process at the same time, users can observe large variability in the response times of particular operations. Users familiar with single process software products (such as native NX), which show much less variation, may not be accustomed to such variation, but may notice inconsistent response times for certain Teamcenter operations, such as loading an assembly into NX from Teamcenter.

In some cases, performance issues are introduced when the system is initially installed and configured. In many however, response times degrade over time, or increase suddenly as the result of some change in the data, environment, or software. Customizations have been observed to noticeably affect performance as well.

10.1.3 ***Monitoring Performance***

The distributed nature of Teamcenter can also exacerbate the process of isolating performance problems. A careful, methodical analysis of the symptoms and response times, together with stringent environment change management procedures, are essential to optimize performance and correct response time problems. Take response time measurements of key Teamcenter operations to serve as a baseline. Periodically measure those operations against the baseline metrics to detect if performance is changing. Once the system is optimized, your goal should be detecting negative performance trends before catastrophic degradation occurs. With each software, hardware, or configuration change, measure response times again to determine the impact, and back the change out if performance becomes unacceptably worse.

Monitor system utilization regularly once the system is in production use. Usage profiles change as users find new methods to accomplish old tasks. Monitor systems regularly to flag potential performance problems before they become critical. Use of a comprehensive system utilization tool from the server platform manufacturer such as OpenView from Hewlett-Packard, Sun Management Center from Sun, or PerfMon on Windows systems, is recommended to monitor the system load on the machine hosting the Teamcenter Rich Client Servers.

Windows Task Manager provides rudimentary information on process execution, memory usage, handles and threads, and a number of other metrics. The Microsoft Performance Monitor (PerfMon) is preferred as it can provide rather detailed information for the overall system, memory utilization, disk and network I/O, specific instances of processes, and numerous other statistics.

HP OpenView and Sun Management Center have the added advantage of being able to trigger email or pager notification if system resources reach critical levels. HP OpenView can also be directly integrated with ARM compliant performance instrumentation in Teamcenter.

Use these tools frequently to monitor the health of system resources. As critical resources approach limits that may affect user response times, supplement the resources as required. This could include:

- Adding or upgrading CPUs or memory on server systems
- Adding disk space or upgrading to faster disks
- Adjusting operating system kernel or configuration parameters
- Splitting users across server systems or reallocating users to different servers
- Moving the location of servers to be closer on the network to heavy users
- Rerouting network connection along different paths or adding network interface cards (NICs) to server machines.

Taking these precautionary measures may help eliminate excessive delays that cause user productivity to suffer.

10.1.4 *Tracking Performance and Analyzing Symptoms*

One or more system monitoring tools will be needed to track and analyze system performance. These tools examine system utilization and load and help you diagnose the cause of performance problems. Such tools are available from most operating system and platform suppliers (such as Microsoft, Hewlett-Packard, and Sun Microsystems) and provide varying degrees of coverage and features. Some examples include the Windows Performance Monitor (Perfmon), HP OpenView, and the Sun Management Center. Basic tools are often provided with the operating system (for example, Microsoft Performance Monitor). Oracle also provides management and performance monitoring tools for its database products, as does IBM for DB2 and Microsoft for SQL Server.

Very capable third-party versions are also available at varying costs. For example, Process Explorer, available from Microsoft *Sysinternals Suite*, is a free downloadable utility that can also provide summary performance information as well as process specific details, or *TaskInfo*, available at *http://www.iarsn.com/*, another downloadable operating system monitoring tool for Microsoft Windows.

More sophisticated tools provide alarm capabilities that can notify operations personnel if a predefined event occurs or system capacity is exceeded. On the high end, a broad line of more costly operating system and database monitoring tools are available from *Quest Software*, *BMC Software*, and others.

10.2 Performance Sensitive System Components

The chapter on *Teamcenter Architecture* located in the *Teamcenter Help Library* includes a description of Teamcenter software components. (See *Site Planning, Installing Teamcenter components* in the *Installation on UNIX and Linux Servers* or *Installation on Windows Servers* books under *Installation* on the *Online Help Library* main page.)

Those components sensitive to performance issues include:

- Database (Oracle, SQL Server, DB2)
- Web Application server (such as WebLogic, WebSphere, etc.)
- Enterprise server (Server Pool Manager – SPM, or business logic server)
- FMS/TCFS (typically executes on the volume server)

NOTE

Specific performance tuning advice for individual Teamcenter components (Teamcenter Web, Enterprise, and FMS/volume servers, as well as NX/Manager) is now included in the corresponding chapters.

Performance-sensitive system components also include hardware and network components and configurations, including:

- Server CPU, memory, and disk throughput (quantity and speed)
- Client CPU, memory, and disk throughput
- Operating system (OS) configuration and/or kernel/registry settings
- Network capacity (speed, latency, throughput)

There are also a number of performance related Teamcenter configuration options, preferences, and environment variables that can affect performance. A few examples include:

- NX/Manager SQL Optimization
- FMS File Server Caches (FSCs) and File Client Caches (FCCs)

Experience shows that the most severe Teamcenter performance issues are caused by:

14) Database configuration
15) Inadequate (slow or undersized) computer hardware
16) Network latency and throughput

The next few subsections address computer hardware, related operating system issues, and network throughput. The concepts therein and the available remedies are relatively limited, but can have a noticeable impact on performance. So although not much space is devoted to this information, be certain to first ensure that inadequate or incorrectly configured hardware is not at the root of existing performance issues.

10.2.1 *Computer Capacity*

Ensuring that server and client computer systems have adequate capacity is the primary consideration for all Teamcenter performance issues. Teamcenter systems are sensitive to the following computer system resources shown in Table 10-1.

Table 10-1, Teamcenter Resource Sensitivity (High, Medium, Low)

	Servers					Clients	
Resource Category	Database	Enterprise	Web	Volume	FMS	Rich	Thin
CPU integer throughout[75]	H-M	H-M	M-L	L	L	**H**	M
Memory performance	**H**	H	L	L	L	**H**	M
Disk throughput	**H**	L	L	**H**	L	L	L
Network throughput	**H**	**H**	H	**H**	**H**	H	L
Network latency[76]	**H**	**H**	M	**H**	M	**H**	**H**

As Table 10-1 shows, database servers are highly sensitive to all resource categories. Volume servers, including FMS, are less sensitive to CPU but are more so to disk and network throughput. The sensitivity of Enterprise and Web servers varies depending on usage, but network latency (to the database server) and memory availability tend to be Enterprise server bottlenecks with high numbers of users or high transaction rates. Rich clients are very susceptible to CPU and memory, as well as network latency, while Thin (browser) Clients are more tolerant of CPU and network limitations.

Rich clients can take significant CPU and memory resource, depending on the tasks being performed. The Rich client, being a Java application, relies on the underlying Java Runtime Environment (JRE), and is therefore sensitive to the performance of the JRE, which varies from platform to platform.

NOTE

Although CAD systems like NX requires a significant amount of CPU floating-point resource, most Teamcenter components stress the integer throughput capabilities of a system and are not as susceptible to floating-point performance. When you evaluate server systems for Teamcenter, consider the throughput performance. A useful source of information for the throughput capabilities of various platforms is SPEC, the Standard Performance Evaluation Corporation. SPEC publishes a variety of standardized performance statistics including SPECint_rate2000/2006, an integer throughput benchmark. Submitted results in the SPEC CPU2000/2006 category are available at the following URL:

http://www.spec.org/

[75] Keep in mind that throughput is a separate consideration from raw CPU speed.

[76] Ping time

10.2.2 ***Resource Capacity Impact***

The following sections briefly consider these four resources (CPU, memory, disk, network) individually. Keep in mind that resources have an impact on each other. For example, insufficient memory may cause excessive paging to disk, which will increase CPU utilization and can cause disk utilization to exceed its capacity. Always examine all system resources during performance analysis.

See also section 4.5.4, *Impact of Overloading Enterprise Server CPU* for additional information about the impact overloading system resources can have on Teamcenter response times.

10.2.2.1 CPU Resources

For nearly all applications, CPU utilization should be kept below 80–90 percent. Above that level, response times and throughput degrade more rapidly. If CPU utilization is above 80%, and the CPU Process Queue[77] exceeds 2x the number of CPUs, you have a CPU bottleneck[78]. Consider additional and/or faster CPUs to relieve this condition.

10.2.2.2 Memory Resources

All applications perform better if their programs and data are always available in physical memory. Memory should therefore always be sized to minimize paging to disk for performance critical applications. Check the total memory utilization (percent) and virtual memory (VM) page-in and page-out rates. Ideally, nearly 100 percent, but not all, of memory should be in use (otherwise, it is wasted). But, if total utilization is near 100 percent and either VM activity is high or page-out activity is high, your system may be experiencing memory pressure. Paging increases disk I/O and can rapidly decrease application response time.

10.2.2.3 Disk Resources

Nearly all Teamcenter applications and processes manage data in memory structures as much as possible. So unless there is memory pressure that results in disk paging, there is relatively little disk activity. Except for database and volume servers (and potentially custom written applications) there should be virtually no disk activity once the application has been started.

Teamcenter applications use dynamic, shared libraries wherever possible. These libraries are used by all instances of applications that need them, and are normally loaded only

---

[77] CPU Process Queue is the number of processes that are waiting for the CPU to resume execution. This does not include processes that are executing, waiting (memory, I/O, streams), or sleeping.

[78] There are additional constraints that indicate a CPU bottleneck that vary with platform. Consult your platform supplier for more information about how to detect and alleviate CPU capacity issues.

when needed. Therefore, there may be occasional disk activity as application features are used that require a library for the first time.

Optimizing database disk performance and configuration, including RAID selection, is presented in chapter 5, *Managing Database Server Deployments*.

10.2.2.4 Network Resources

The network interconnecting Teamcenter components is one of the most critical and often overlooked elements affecting performance. All data presented to or generated by the application must traverse the network. It is a misconception that network speed alone is the limiting factor when considering performance issues. There are actually three factors that affect network throughput:

- Speed of the medium (bits per second)
- Distance between systems (latency or propagation delay)
- Utilization of the line (congestion)

Network Bandwidth

Speed of the medium varies by technology. Copper wire based medium now installed in many facilities support 100-Mbit transmission speeds. One Gbit transmission speeds are also available, but require fiber-optic cables or special copper cables.

NOTE

Siemens PLM Software recommends a 100-megabit (Mbit) network interface (NIC) as a minimum for all Teamcenter client applications.

One gigabit (Gbit) is recommended for all Teamcenter servers supporting more than 100 users.

Network Latency

The distance between systems also impacts end-to-end response time if the propagation delay exceeds more than a few milliseconds (ms). Performance is noticeably slower if the propagation delay exceeds 10 ms. Latency is the round-trip propagation delay and can be measured with **ping** or a network monitoring/analysis tool. Most local area networks (LANs) in a single building exhibit latencies under 2–5 ms (actually most are < 1 ms).

Latency is mostly determined by distance, but can also be impacted by internodal delay. Internodal delay is the time it takes for a network switch or gateway to retransmit the data from one network segment to another, or *hop* segments. So in a typical wide area network (WAN) path with multiple gateways, the *hops* can also contribute to latency. Fewer hops can reduce latency a small amount. Land-based WANs can exceed 300 ms, while satellite-based connections approach one second.

For optimal performance, Siemens PLM Software recommends all Teamcenter components except clients reside on the same LAN with less than 10 ms latency. Thin Clients are less sensitive to network latency, so if remote users must access the system over very high latency, consider the Teamcenter Thin Client for those users wherever possible. Under no circumstances should satellite-based WANs be used for Teamcenter applications.

Utilization

Teamcenter performance is also impacted by high network utilization. Congested networks require the application to wait and can cause information to be resent multiple times. Network congestion can be indicated with a sharp rise in network errors (collisions, alignment errors, framing errors, and so forth).

One can examine network errors with a variety of platform performance monitoring tools, or your network hardware may provide monitoring capabilities. If the network appears congested, consult with the network administrator to consider system upgrades or topological changes to reduce network congestion.

Field experience has shown that the suitability of operating over a WAN varies according to several general considerations such as cost, processes, configuration, and applications. **Table 10-2** provides some guidelines as to the suitability of Teamcenter deployment options for various general considerations.

Table 10-2, WAN Deployment Suitability

General Categories	Sub Categories	4-tier, no local FSC	4-tier, local FSC	4-tier local Volume	4-tier S&F
Cost	Administration	Very Low	Low	Medium	Medium
	Hardware	Very Low	Medium	Medium	Medium
	Software	Very Low	Low	Low	Low
Business Processes	Frequent Data Sharing	Perfect Fit	Perfect Fit	Perfect Fit	Perfect Fit
	Transfer Of Ownership of CAD	Perfect Fit	Perfect Fit	Perfect Fit	Perfect Fit
	High Level Interaction (e.g. follow the sun)	Perfect Fit	Perfect Fit	Perfect Fit	Perfect Fit
	Security (data partitioning)	Supported	Supported	Supported	Supported
	Managed Environment (for CAD)	Perfect Fit	Perfect Fit	Perfect Fit	Perfect Fit
Environment	1-25 users	Good Fit	Perfect Fit	Perfect Fit	Perfect Fit
	25-100 users	Not a strong fit	Good Fit	Perfect Fit	Perfect Fit
	101-500 users	Not a strong fit	Good Fit	Good Fit	Good Fit
	>500 users	Not a strong fit	Good Fit	Good Fit	Good Fit
	Unreliable Network	ill suited	ill suited	ill suited	ill suited
	Latency: 0-20ms	High	High	High	High

Table 10-2, WAN Deployment Suitability (cont.)

General Categories	Sub Categories	4-tier, no local FSC	4-tier, local FSC	4-tier local Volume	4-tier S&F
Local CAD Performance	Latency: 20-100ms	Acceptable	Acceptable	Acceptable	Acceptable
	Latency: 100-200ms	Acceptable	Acceptable	Acceptable	Acceptable
	Latency: 200-300ms	Low	Low	Low	Low
	Latency: >300ms	Not Supported	Not Supported	Not Supported	Not Supported
	Bandwidth: 1Gbps +	High	High	High	High
	Bandwidth: 100Mbps	Acceptable	High	High	High
	Bandwidth: 10-100Mbps	Acceptable	High	High	High
	Bandwidth: 1-10Mps	Low- < 10 clients	Acceptable	High	High
	Bandwidth: <1 Mbps	Very Low	Poor < 5	Acceptable < 5	Acceptable < 5
Local Visualization Performance	Latency: 0-20ms	High	High	High	High
	Latency: 20-100ms	Acceptable	Acceptable	Acceptable	Acceptable
	Latency: 100-200ms	Acceptable	Acceptable	Acceptable	Acceptable
	Latency: 200-300ms	Low	Low	Low	Low
	Latency: >300ms	Not supported	Not supported	Not supported	Not supported
	Bandwidth: 1Gbps +	High	High	High	High
	Bandwidth: 100Mbps	Acceptable	High	High	High
	Bandwidth: 10-100Mbps	Acceptable	Acceptable	Acceptable	Acceptable
	Bandwidth: 1-10Mps	Low- < 10 clients	Acceptable	Acceptable	Acceptable
	Bandwidth: <1 Mbps	Poor - < 3 clients	Poor - < 3 clients	Poor - < 3 clients	Poor - < 3 clients
Client Core function Performance	Latency: 0-20ms	High	High	High	High
	Latency: 20-100ms	Acceptable	Acceptable	Acceptable	Acceptable
	Latency: 100-200ms	Acceptable	Acceptable	Acceptable	Acceptable
	Latency: 200-300ms	Acceptable	Acceptable	Acceptable	Acceptable
	Latency: >300ms	Not supported	Not supported	Not supported	Not supported
	Bandwidth: 1Gbps +	High	High	High	High
	Bandwidth: 100Mbps	Acceptable	High	High	High
	Bandwidth: 10-100Mbps	Acceptable	Acceptable	Acceptable	Acceptable
	Bandwidth: 1-10Mps	Acceptable	Acceptable	Acceptable	Acceptable
	Bandwidth: <1 Mbps	Acceptable - < 3 clients	Acceptable - < 3 clients	Acceptable - < 3 clients	Acceptable - < 3 clients

An excellent document with guidelines for Network Performance Tuning is available on the *Global Technical Access Center* (*GTAC*) Teamcenter documentation page. This document also includes information about improving WAN performance with network acceleration devices such those from Riverbed, Cisco and Blue Coat.

10.2.3 *Operating System Considerations*

In many instances, operating system parameters can be adjusted, or *tuned* to provide better performance. Some parameters select alternate algorithms optimized for certain types of applications (for example, visualization intensive vs. compute intensive). Others allocate the appropriate level of resources to support more or fewer users (for example, file descriptor table entries).

UNIX operating systems expose numerous kernel parameters for this purpose. These are altered either in text files or with tools provided by the operating system provider. Although providing flexibility, adjusting OS kernel parameters can prove daunting to the novice system administrator. Setting parameters improperly can render the system virtually inoperable, or prevent some applications from running. Consult your operating system administration manual for guidance.

Microsoft Windows manages operating system parameters, as well as numerous application parameters in the *system registry*. The registry can be altered only with the **regedit.exe** tool. Siemens PLM Software recommends the registry be modified only by a certified Windows system administrator, or when explicit instructions are provided. Even with explicit instructions, consult an experienced Windows system administrator before making changes if the contents of the registry do not agree with the instructions; registry structure can change from Windows version to version.

Also note that hardware suppliers often publish operating system performance tips and/or guidelines that provide useful configuration and tuning guidelines; many available on their Web sites.

10.2.3.1 Suse

Transparent Huge Pages (THP) Support

On Linux systems with large memory, frequent access to the Translation Lookaside Buffer (TLB) may slow down the system significantly.

Transparent huge pages thus are of most use on systems with very large (128GB or more) memory, and help to drive performance. In SUSE Linux Enterprise, THP is enabled by default where it is expected to give a performance boost to a large number of workloads.

There are cases where THP may regress performance, particularly when under memory pressure due to pages being reclaimed in an effort to promote to huge pages. It is also possible that performance will suffer on CPUs with a limited number of huge page TLB entries for workloads that sparsely reference large amounts of memory. If necessary, THP can be disabled via the sysfs file "/sys/kernel/mm/transparent_hugepage/enabled", which accepts one of the values "always", "madvise", or "never".

To disable THP via sysfs and confirm it is disabled, do the following as root:

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
```

10.2.4 *Customizations*

Customizations involving programming in the Rich Client using Java and the Eclipse IDE, ITK, or user exits can have a significant impact on performance. Wherever possible, structure custom code so that it can be easily enabled and disabled. While this may not always be possible, doing so makes it easier to isolate performance issues introduced by your customizations

Glossary

This appendix defines selected Teamcenter terms.

C

Client Tier

Teamcenter architectural tier that comprises the Teamcenter clients, Teamcenter integrations with third-party applications, such as Teamcenter's Integration for Microsoft Office and AutoCAD and the third-party applications themselves such as Microsoft Office and AutoCAD.

Corporate Server

Host computer at the center of a Teamcenter network. This host contains the Teamcenter application root directory, Teamcenter data directory, licensing, file managers (Teamcenter File Services and File Management System), and volumes. For installations that include the Web tier (four-tier architecture), the corporate server also contains the Teamcenter server manager. Multiple application clients can map to or mount the corporate server.

E

Enterprise Archive (EAR)

Enterprise application that requires a J2EE application server

Enterprise Tier

Teamcenter architectural tier that comprises a configurable pool of Teamcenter C++ server processes and a server manager. Larger sites can distribute the pool of server processes across multiple hosts. Smaller sites can run the pool of servers on the same host as the Web tier.

F

File Management System (FMS)

System that manages uploading and downloading file data between clients and volumes in both two-tier and four-tier architecture deployments. FMS provides volume servers for file management, a shared server-level performance cache for shared data access between multiple users, a client-based private user cache for rich clients, and a transient datastore mechanism for transporting reports, PLM XML, and other non-volume data between the enterprise and client tiers. FMS file caching enables placing the data close to the user, while maintaining a central file volume and database store

Four-Tier Architecture

Teamcenter architecture that includes four tiers: resource tier, client tier, Web tier, and enterprise tier.

Four-Tier Deployment

Deployment of the Teamcenter four-tier architecture. The Web tier, enterprise tier, resource tier, and client tier can each be hosted on the same or separate computers.

H

Heterogeneous Network Environment

Teamcenter network environment in which the database server, Teamcenter application and volume servers, *TC_DATA* server, and Teamcenter clients are provided by multiple vendors and can be a mixture of UNIX and Windows nodes.

Compare with *Homogeneous Network Environment*

Homogeneous Network Environment

Teamcenter network environment in which the database server, Teamcenter application and volume servers, *TC_DATA* server, and all Teamcenter clients are provided by one vendor (for example, Sun Microsystems).

Compare with *Heterogeneous Network Environment*

I

IDSM Server

Teamcenter Distributed Services Manager. A network node that runs a daemon process to handle the transfer of data objects among sites (databases) within a Multisite environment. One IDSM server node must be designated for each Teamcenter database from which objects are published; each server node can act only for one database.

M

Multisite

Application that allows the exchange of data objects among several Teamcenter databases. Transfer of objects among databases is controlled by daemon processes running on designated servers. Objects are replicated by exporting them from their original database and importing them into the requesting database. Configuration of Teamcenter Multisite is optional.

N

NX Integration

Integration between Teamcenter and NX. NX Integration users have full access to the Teamcenter user interface from NX, and they can also access NX from the Teamcenter user interface.

O

Object Directory Services Server

Classic Multi-site Collaboration network node that runs a daemon process to handle publication of data objects within a Classic Multi-site Collaboration environment. One ODS server node must be designated for each object directory services site and each server node can act only for one object directory services site.

Object Directory Services Site

Site with the database that maintains a record of each object in a Classic Multi-site Collaboration network. At least one Teamcenter database on a Classic Multi-site Collaboration network must be designated as an ODS site. This site is used to store publication records for the data objects.

ODS Server

See *Object Directory Services Server*.

ODS Site

See *Object Directory Services Site*.

Oracle Home

Directory in which Oracle software is installed on the Oracle server node. The **ORACLE_HOME** environment variable defines this directory.

Oracle Server

Single installation of Oracle able to service queries from several Teamcenter workstations. The **ORACLE_SERVER** environment variable defines this Oracle service node. For large-scale installations, the Oracle server is typically a dedicated high performance system that is optimized specifically for running Oracle software.

Oracle System Identifier (SID)

Alphanumeric text used to identify a collection of processes and associated memory structures as belonging to a particular Oracle database instance. The **ORACLE_SID** environment variable defines the Teamcenter-Oracle system identifier.

Q

QPL

See *Quick Part Locator*

QPL Server

Quick part locator server. It provides a **QPL** daemon that must be used with Design in Context in Rich Client.

Quick Part Locator

Component of Repeatable Digital Validation/Quick Part Locator (RDV/QPL) that creates and queries tables in an Oracle database. The quick part locator enables rapid lookup of components in an assembly by their location or properties. These tables cache pre-computed results of location, attributes, and spatial relationship of all components in a product assembly.

R

Resource Tier

Teamcenter architectural tier comprising the database Server, database file Servers, and volumes

Rich Client

Java-based user interface to Teamcenter installed on user workstations. The rich client accesses Teamcenter databases using a remote or local server

RDV/QPL

See *Repeatable Digital Validation/Quick Part Locator*

Repeatable Digital Validation/Quick Part Locator

Combination of software and processes that enables on-demand digital mockups for use throughout the product development lifecycle

S

Server Manager

Process that manages a pool of Teamcenter Server processes in a deployment of the four-tier architecture. The Server manager starts and times out a configurable number of Server processes to communicate with the Teamcenter database. A Server assigner process assigns available Server processes to user sessions. The Server manager communicates with the Web tier application using either TCP or multicast protocol.

Server Pool

Pool of Teamcenter Server processes running in the enterprise tier. A small deployment may have only one pool of Server processes. For larger deployments, the pool of Server processes is distributed as sub pools across multiple hosts, with a Server manager for each sub pool. Server pools are applicable for deployments of the Teamcenter four-tier architecture only.

T

Teamcenter's lifecycle visualization

Siemens PLM Software suite that provides enterprise-wide product visualization capabilities. Teamcenter's lifecycle visualization can be configured for use with both the Teamcenter rich client and thin client as a stand-alone application.

The software suite includes Teamcenter's lifecycle visualization base, Teamcenter's lifecycle visualization standard, Teamcenter's lifecycle visualization professional, and Teamcenter's lifecycle visualization mockup.

Teamcenter Application Root Directory

Directory location of the Teamcenter shared binary executables. The **TC_ROOT** environment variable defines this location. Generally, the contents of this directory change only with a new version of Teamcenter.

Teamcenter Data Directory

Directory location of the Teamcenter shared data subdirectories and files. The **TC_DATA** environment variable defines this location. Each data directory is associated with a single database instance

Teamcenter Engineering Visualization

Optional Teamcenter rich client component that provides enterprise-wide product visualization capabilities and is embedded in the rich client interface. Four Teamcenter Engineering Visualization products provide different visualization capabilities. Teamcenter Engineering Visualization Base provides basic 2D and 3D viewing. Teamcenter Engineering Visualization Standard, Teamcenter Engineering Visualization Base, and Teamcenter Engineering Professional provide increasing functionality.

Teamcenter File Services (TCFS)

File services that enable the Organization application to create volumes and perform other administrative functions. TCFS also supports file access for legacy versions of NX and Teamcenter's lifecycle visualization.

Teamcenter File Structure

File structure of an installed Teamcenter node; it separates homogeneously shared binary executables from heterogeneously shared data.

Thin Client

Teamcenter user interface that provides a streamlined browser-based view of product information stored in a Teamcenter database.

Transient Volume

Operating system directory controlled by Teamcenter and used to store temporary data for transport of reports, PLM XML, and other non-volume data between the Web tier and client tier in a deployment of the Teamcenter four-tier architecture.

Two-Tier Architecture

Teamcenter architecture that includes a resource tier and a client tier. The resource tier comprises the database Server and database. The client tier comprises the Teamcenter rich client, third-party applications that integrate with the rich client, and a local Server. This architecture supports only the Teamcenter rich client.

Two-Tier Deployment

Deployment of the Teamcenter two-tier architecture. In a typical deployment of the two-tier architecture, the rich client and its local Server are installed on a user's workstation as are third-party applications that integrate with the rich client. The database Server and the Teamcenter corporate Server are installed on one or more separate computers.

V

Volume

Operating system directory controlled by Teamcenter and used to store the files managed by Teamcenter. When a user performs an action that causes Teamcenter to create a file, the file is created in the Teamcenter volume. Users cannot directly access the files in Teamcenter volumes; they must do so via a Teamcenter session.

W

Web Application Manager

Graphical installation utility that generates supporting Web files (WAR and EAR format) for a named Web application. Web Application Manager also installs the rich client distribution Server and creates distribution Server instances.

Web Tier

Teamcenter architectural tier that comprises a Java application running in a Java 2 Enterprise Edition (J2EE) application Server. The Web tier is responsible for communication between the client tier and enterprise tier. The Web tier application also includes the Application Interface Web Service (AIWS), WebDAV service, and thin client.

Appendix

A *Sample User Scenarios*

This appendix provides user sample scenarios for estimating Teamcenter sizing requirements.

The following scenarios were derived from the Thin Client User Interface and can be used as examples to help define a Teamcenter *Usage Profile*. The *Usage Profile* itself (that is, the number of these scenarios performed per day by each type of user) is defined separately in appendix B, *Sample Usage Profiles*.

The scenarios consist of three types of users (Data Analysis, Data Review, and Documentation). The scenario was designed to simulate a 4-5 hour workday.

Scenario #1	**Data Analysis Users**
Overview	The Data Analysis reviews Product Structures at a rate of 2 per hour. This equates to 16 per day in a normal 8-hour shift. The user performs the following tasks to accomplish a single review.
Activities	1. Start Web 2. Enter username and password then click the Log In button 3. Click the HOME icon if you didn't just login. 4. Submit a query based on randomized value of BomID (LVLS_2_LINES_100, etc...). 5. Select View -> Properties from the menu bar. 6. Click on the HOME icon next. 7. Submit a query based on randomized value of BomID (LVLS_2_LINES_100, etc...). 8. Click the LVLS_x_LINES_yyyy link to view the bom. 9. Do research, take a break or whatever and repeat this again later. 10. It may be time to go home. If so, click the Logout icon
Users	• user_an0 - user_an99 (100) • user_pe0 - user_pe99 (100) • user_pd50 - user_pd99 (50)

Scenario #2	Data Review Users
Overview	The Data Review user performs 2 reviews per hour. This equates to 16 per day in a normal 8-hour shift. The user performs the following tasks to accomplish a single review. The user waits approximately 3 seconds in between tasks (think time).
Activities	1. Start Web 2. Enter username and password then click the Log In button 3. Click on the Green InBox icon. 4. Look for any jobs, if found and is 1st iteration, stop -- database must be corrupt otherwise fail because initiate probably didn't work previously. 5. Enter a dataset id such as user_me0_53348 and click on the Go button. 6. Verify the search returned a result, if not we failed. 7. Click the Dataset ID hyperlink under the Name column. 8. Click on the UGMASTER link next. Take a second or two to find it. 9. Click the HOME icon. 10. Do another search for the same dataset id. 11. Select the View -> Where Referenced menu option. 12. Clicking on the HOME icon again. 13. Once again, search for the same dataset id. 14. Select the View -> Where Used (All Revisions) option. 15. Click the HOME icon; we plan to create a new form next. 16. Search for the same dataset so we can attach a form to it. 17. Click on the item next. 18. Select New -> Form... Name=Impact Analysis of <dataset>, Description=LR Generated 19. Click on the UGMASTER next. 20. Select Action -> Initiate New Process 21. Complete New Job form: - Job=Work Authorization for 53348 - Description=LR Initiated - Process=CMII WA 22. Click the InBox icon next 23. Verify that no other job but Master Control Document exist, if there are we failed. 24. Start by clicking on the Master Document Control link. 25. Mark as done. 26. Make sure we got the Work Copies task, if not we failed.

Scenario #2	Data Review Users
	27. Complete Work Copies next. 28. Mark as done. 29. Complete Prepare Work Authorization next. 30. Mark as done. 31. Complete Physical Items next. 32. Mark as done. 33. Do research, take a break or whatever and repeat this again later. 34. It may be time to go home. If so, click the Logout icon
Users	• user_me0 - user_me99 (100) • user_pl0 - user_pl99 (100) • user_pd0 - user_pc49 (50)

Scenario #3	Documentation Users
Overview	The Documentation user creates 1 folder, item and text dataset then revises it per hour. This equates to 8 per day in a normal 8-hour shift. The user waits approximately 3 seconds in between tasks (think time).
Activities	1. Start Web 2. Enter username and password then click the Log In button 3. If the LR_Text_mmddyy folder doesn't exist yet, create it. 4. Click the LR_Text_mmddyy hyperlink to open this folder. 5. Select New -> Item... from the menu. - Item ID: LR_Item_<User-Name>_<Create-Time> - Item Name: LR_Item_<User-Name>_<Create-Time> - Item Description: Generated by LoadRunner 6. Click the items' hyperlink to open the item to attach a text dataset. 7. Select the New -> Dataset from the menu bar. 8. Click on the link to get to the upload dialog. 9. Enter filename into upload dialog. 10. Click the submit button. 11. Select the box next to the dataset then select Edit -> SaveAs 12. Change description to 'Revisioned by LoadRunner' 13. Locate and click the Home icon. 14. Click the LR_Text_mmddyy hyperlink to open this folder. 15. Click the items' hyperlink to open the item again. 16. Locate and click on the .jpg object to view it. 17. Select the box next to the text dataset. 18. Select from the menu, dataset then Actions -> Check out. 19. Select from the menu, Actions -> Cancel Checkout. 20. Do research, take a break or whatever and repeat this again later. 21. It may be time to go home. If so, click the Logout icon
Users	• user_br0 - user_br99 (100) • user_cd0 - user_cd99 (100) • user_sd0 - user_sd99 (100) • user_qa0 - user_qa99 (100) • user_sa0 - user_sa99 (100)

Appendix

B Sample Usage Profiles

This appendix provides an example *Usage Profiles* similar to those used in the Teamcenter Rich Client scalability benchmarks to help qualify system sizing.

Typical Usage Profiles

A typical usage profile will characterize various types of users, often by the role they fill in your company. There are likely varying numbers of each user type, and each type generally execute different Teamcenter operations depending on their role. Of course there numerous operations that all users exercise to some extent or another, such as login, open folder, create item, etc.

When planning a new Teamcenter deployment, be sure to interview as many end users as practical and document the various user types, the estimated count of each type, the Teamcenter operations they use most, and approximately how often each operation is used in a day. If the new Teamcenter deployment is replacing an existing legacy system, try to obtain empirical usage data from the legacy system to help estimate the amount of load that might be placed on the Teamcenter deployment.

In the example below, five user types are shown along with the percentage of the workforce that type represents, the scenarios they typically execute, and how many times in an 8 hour day they perform them.

Table B-1, Sample Usage Profile – Rich Client

Profile/%: **Reviewer** - 50%
Role: Reviewer
Group: Engineering
Project: N/A

- Login
  - 5 Iterations
    - View Item Properties
    - View BOM
    - View Image
    - Create Folder
    - Display Designer Reports
    - Copy Item
    - Paste Item
    - Cut Item
    - Delete Folder
    - Expand Below PSE
    - Search F (Feature/Function)
    - View F Properties
    - SE Expand Below – 1 Level F
    - Expand Relations - Traceability
    - Search L (Logical Element)
    - View L Properties
    - SE Expand Below – 1 Level L
    - Expand Relations - Traceability
  - next iteration
- Logout

Review RFLP Structures - Frequency lower than other SE Core actions

Profile/%: **RM Author** - 10%
Role: Requirements Author
Group: Engineering
Project: N/A

- Login
  - 2 iterations
    - Create Requirement
    - View Requirement
    - View Requirement Properties
    - Save Requirement Properties
    - Quick Release Requirement
  - next iteration
- Logout

Manage Requirements

Profile/%: **SE Core Once Daily** - Once Per Day
Role: Systems Engineer - Core
Group: Group 1
Project: Project 1

- Login
  - 1 iterations
    - SE Duplicate 2nd level R (200 BOM)
    - SE Duplicate 2nd level F (1000 BOM)
    - SE Duplicate 2nd level L (320 BOM)
  - next iteration
- Logout

Clone Commodity RFL Structures - Frequency lower than other SE Core actions

Profile/%: **SE Core** - 10%
Role: Systems Engineer - Core
Group: Group 1
Project: Project 1

- Login
  - 1 iterations
    - SE Expand Below All - 2nd level R
    - SE Expand Below All - 2nd level F
    - SE Expand Below All - 2nd level L
    - Create F (Feature/Function) Item
    - Create and Attach Form
    - Create and Attach Dataset
    - Create L (Logical Element) Item
    - Create and Attach Form
    - Create and Attach Dataset
    - Create FL Relation
    - Search R (Requirement)
    - Create RF Relation
    - Expand Relations - Traceability
    - Search P (Physical)
    - Create LP Relation
    - Expand Relations - Traceability
    - Revise F (Feature/Function)
    - Revise L (Logical Element)
    - Delete Form
    - Delete Dataset
  - next iteration
- Logout

Create/revise RFL items, create/update structures

Profile/% : **SE App** - 30%
Role: Systems Engineer - App
Group: Group 1
Project: Project 1

- Login
  - 5 iterations
    - Search F (Feature/Function)
    - SE Expand Below – 1 Level F
    - Create and Attach Form
    - Create and Attach Dataset
    - Search L (Logical Element)
    - SE Expand Below – 1 Level L
    - Create and Attach Form
    - Create and Attach Dataset
    - Create FL Relation
    - Expand Relations - Traceability
    - Search Requirement
    - Create RF Relation
    - Expand Relations - Traceability
    - Search P (Physical)
    - View JT
    - Create LP Relation
    - Expand Relations - Traceability
    - Delete Form
    - Delete Dataset
  - next iteration
- Logout

Create relationships and update existing objects – no new items/revs/structures

Appendix

C Teamcenter Operational Maintenance

This appendix lists utilities that can be used for the operational maintenance of the system along with recommended frequency and potential impact to users and data.

Table C-1 lists Teamcenter utilities that can be used for the operational maintenance of the system. This is not an exhaustive list of all of the utilities, but rather those functions that should be performed on a scheduled basis. This list should be considered for inclusion into the standard operational processes of the data center.

Table C-1, Operational Maintenance Utilities

Utility Function	***Impact***	***Frequency***	***Notes***
list_users Create a list of users currently logged in and what node they are using.	None	As needed	Useful if there is a planned maintenance or to snapshot a user count for trending.
clearlocks Clear application locks held by improperly terminated Teamcenter processes.	Low Provided that assert_dead or assert_all_dead arguments are not used.	Weekly	Application locks can consume database resources and can affect user performance.

Table C-1, Operational Maintenance Utilities

Utility Function	***Impact***	***Frequency***	***Notes***
dataset_cleanup Repairs corrupted datasets and removes orphaned revision anchors.	Medium	Weekly Performed in the off hours.	A dataset is identified as corrupted if any of the following problems are found: Dataset has no reference to a **TcFile** object. Dataset has reference to a **TcFile** object, but the corresponding operating system file does not exist and the dataset is not archived. Dataset is an orphan (that is, the dataset refers to the anchor but the anchor does not go to dataset). Anchor refers to datasets that do not exist. Anchor size = 0.
index_verifier Detects missing indexes in a Teamcenter database.	Low Can be run against the database at any time.	Monthly or after a product change/upgrade	The results of this utility can help to improve database performance There are five types of indexes that can be detected using this utility: Indexes on the primary key of each Teamcenter class Indexes on variable length arrays (VLA). Indexes created by Teamcenter Functional indexes. This utility detects the necessary functional indexes required by the version of Teamcenter in use. Indexes on system tables such as pom_backpointer, pom_m_lock, and the pm_process_list tables.

Table C-1, Operational Maintenance Utilities

Utility Function	***Impact***	***Frequency***	***Notes***
purge_datasets Removes (purges) old versions of datasets from the database.	Medium	Weekly during maintenance hours	Normally, Teamcenter stores a fixed number of dataset versions in the database. The maximum number of datasets retained is set using the **AE_dataset_default_keep_limit** preference. The purge_datasets utility can be run during normal working hours, with users logged in to the system and can process approximately 500 anchors per hour. A listing is produced that shows each dataset purged, along with the owning user and group.
purge_volumes Removes (purges) operating system files that represent deleted Teamcenter objects.	Low	Weekly during maintenance hours	During a Teamcenter session users delete objects. However, if the user deleting the object does not have the necessary operating system privileges to delete the associated operating system file along with the Teamcenter object, the file remains in the Teamcenter volume. The purge_volumes utility unlocks these files so that they can be deleted at the operating system level
report_volume Lists the operating system path of all existing Teamcenter volumes.	None	Quarterly Or after any version modifications or changes in user storage.	The path does not include the network node name. Use this utility to report on the available volumes and as part of the general housekeeping. Use to create a checklist of volumes for the users.

Table C-1, Operational Maintenance Utilities

Utility Function	***Impact***	***Frequency***	***Notes***
review_volumes Allows you to view detailed information about Teamcenter volumes and to remove unreferenced operating system files from these volumes.	Medium	Weekly during maintenance hours	This utility can generate a report file describing volume usage by various groups and users, as well as reporting any unreferenced operating system files, missing operating system files, and unreferenced Teamcenter files. Unreferenced operating system files can be deleted at the time a report file is generated or at a later time using a previously-generated report file as an input. The report file format is plain text (ASCII) and can be manually edited in order to not delete certain files. Simply remove any file names that you do not want to delete before using the report file as input. You can also save any deleted files to a ZIP format compressed file.
backup_modes Allows Teamcenter to operate in 24 x 7 mode.	High If users are writing files to the volumes, they will be denied access and may experience extended wait times.	Weekly during maintenance hours	This utility administers hot backup so that Teamcenter need not be shut down for routine backup. The **backup_modes** utility sets TcFs (Teamcenter volumes) in different modes and sends appropriate messages to all current Teamcenter sessions. Third-party storage management systems can use this utility to obtain the hot backup of Oracle database and the file system.

Table C-1, Operational Maintenance Utilities

Utility Function	***Impact***	***Frequency***	***Notes***
cleanup_recovery_table Locates and removes all recipe objects resulting from time-out sessions and client crashes.	Medium	Monthly	Teamcenter recipe objects are used to facilitate transparent recovery when a tcserver crashes. Typically, recipe objects are deleted when users log off. But during client crashes or session time-outs, recipe objects may remain in the database if the session or objects are not recovered. Run this utility periodically to clear the database of recipe objects. See the *Teamcenter System Administration Guide* for more information.

Table Key

Utility = Name of the Utility that can be run from the command line or script

Function = What the utility does

Impact = What is the impact (risk) level to the users and data (None, Low, Medium, High)

- **None** No impact, information reporting, can be run at any time
- **Low** May take a short period of time to complete, modifies data
- **Medium** Takes a longer period of time to complete, may affect user performance
- **High** Should only be run when users are not active; DB maintenance and volume maintenance.

Frequency = How often is should be run or scheduled

Notes = Why you want to use the utility and possibly under what circumstances.

Reference: Teamcenter On-Line help. Utilities Reference Guide. Before using ANY of these utilities it is important to read the reference manual for the usage types.

Index

P

Q

R

S

T

U

V

W