Oracle® Agile Product Lifecycle Management for Process

Capacity Planning Guide

Release 6.1.1

E29788-01

December 2014

ORACLE®

Oracle Agile Product Lifecycle Management for Process Capacity Planning Guide, Release 6.1.1

E29788-01

Contents

Preface

The *Agile Product Lifecycle Management for Process Capacity Planning Guide* provides guidelines and recommendation for planning your implementation of Agile Product Lifecycle Management (PLM) for Process.

This preface contains these topics:

Audience

This guide is intended for end users who are responsible for creating and managing information in Agile PLM for Process. Information about administering the system resides in the *Agile Product Lifecycle Management for Process Administrator User Guide*.

Variability of Installations

Descriptions and illustrations of the Agile PLM for Process user interface included in this manual may not match your installation. The user interface of Agile PLM for Process applications and the features included can vary greatly depending on such variables as:

- Which applications your organization has purchased and installed
- Configuration settings that may turn features off or on
- Customization specific to your organization
- Security settings as they apply to the system and your user account

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Software Availability

Oracle Software Delivery Cloud (OSDC) provides the latest copy of the core software. Note the core software does not include all patches and hot fixes. Access OSDC at:

http://edelivery.oracle.com.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Reviewing Components and Requirements

Below is a list of the components of the Agile PLM for Process application suite and directly related third party hardware devices.

Components

Application Server

The Agile PLM for Process Application Server is the base for the PLM for Process platform, where all common services and business logic reside for the entire solution. All client servers and users connect to the Application Server either directly or indirectly.

Remoting Container Server

The Agile PLM for Process Remoting Container Server is a windows service that hosts several application services such as authentication, authorization, reporting, taxonomy denormalization, etc. This service can be hosted on the existing Application Server without any impact to overall performance or scalability.

Database Server

The Agile PLM for Process Database Server persists or stores all product content and system settings. This database can run on supported Microsoft SQL Server or Oracle versions.

Load Balancer

The hardware load balancer brokers client communications without compromising the security of your internal network. Clients communicate through the load balancer with the application server.

There are no Agile PLM for Process software components running on the hardware load balancer. They are usually deployed in the Demilitarized Zone (DMZ) where it proxies requests from outside the corporate firewall to the application server in the Safe Zone.

Reverse Proxy

The reverse proxy acts as an intermediary, retrieving resources on behalf of the client from n number of servers. Typically, a reverse proxy is placed in the DMZ and will proxy client requests from servers in the intranet.

There are no Agile PLM for Process software components running on the reverse proxy, but we do recommend using a reverse proxy for the supplier portal.

Clients

Agile PLM for Process uses a web client; a thin HTML client that uses standard HTTP/S protocols.

Software and Hardware Requirements

For general software and hardware requirements, refer to the Install/Upgrade Guide on OTN for your specific release,

http://www.oracle.com/technetwork/documentation/agile-085940.html#plmprocess

2

Planning Capacity

This chapter includes capacity planning guidelines for the Agile PLM for Process application suite.

Implementation Recommendations

There are a number of factors to consider when determining your implementation size for a production environment, and it is up to the customer's business and IT to determine what makes sense for their organization. However, the simplest method for estimating size is by analyzing the number of planned registered users over the next three years.

Application Server Application Pool Distribution

The following is applicable to all implementation sizes and serves as an initial recommendation for how to distribute your applications in a production environment. With monitoring in place, we recommend you adjust these as needed. The one caveat being, GSM should always reside in its own application pool.

.

Table 2–1 Application pool distribution

Application Pool Name	Applications	Reason
PLM4P_GSM	GSM	GSM is a heavily used application and has the potential to consume the most memory. Therefore, we recommend it should always be isolated in its own application pool.
PLM4P_GSMView	GSMView	GSMView is a read-only copy of GSM with the purpose of off loading certain memory intensive operations. It may not be required to be in an isolated application pool, but should be separated from GSM.
PLM4P_PDM	Integration, ProdikaReporting, SCRM, WFA	
PLM4P_MAIN	Portal, REG, UGM, DRL	
PLM4P_FC	PQS, OPT	
PLM4P_NPD	NPD	NPD is a heavily used application and has the potential to consume more memory than other applications. Therefore, we recommend it should be isolated in its own application pool.
PLM4P_COLLAB	SupplierPortal, SupplierPortalAdmin, eQ	

Basic Recommended Configuration

Based on our test results and sizing and scaling calculations, we have put together a set of recommendations based on implementation size. This does not account for additional servers required for high availability.

These recommendations are based on the number of logged in users, which is different than the number of concurrent users, which is the number of users actively engaged in using the system.

Table 2–2 Recommended configuration

Size	Number of Active Users	Configuration
Small	Less than 200	**Servers**
		1 Internal Application Production with 4 cores and 8 GB RAM
		1 Internal Data Administration Staging with 2 cores and 8 GB RAM
		1 External Supplier Portal/eQ with 4 cores and 8 GB RAM
		1 Internal DB
		Additional Hardware
		Load Balancer (optional)
		Reverse Proxy (optional)
Medium	Between 200 and 1000	**Servers**
		2 Internal Application Production with 4 cores and 12 GB to 16 GB RAM
		1 Internal Data Administration Staging with 4 cores and 8 GB RAM
		1 External Supplier Portal/eQ with 4 cores and 8GB RAM
		1 Internal DB
		Additional Hardware
		Load Balancer (recommended)
		Reverse Proxy (recommended)
Large	Greater than 1000	**Servers**
		2+ Internal Core Application Production with 4 cores and 12GB to 16 GB RAM
		1 Internal Data Administration Staging with 4 cores and 8 GB RAM
		1 External Supplier Portal/eQ (if purchased) with 4 cores and 8 GB RAM
		1 Internal DB (Shared Enterprise)
		1 Reporting DB
		Additional Hardware
		Load Balancer (recommended)
		Reverse Proxy (recommended)

Sizing & Scaling

Application Server

When planning on sizing a production environment, the number of application servers and the amount of memory will need to be determined. The largest factors that determine both of these are the number of users simultaneously logged into the applications as well as the size of the database, especially the amount of custom data.

We are assuming that a 64 bit server is used. A 32 bit server is not recommended because there are significant memory limitations.

Number of Application Servers

Our load testing has determined that a 4 core application server is the optimal configuration and is able to support 50 users per CPU or a total of 200 concurrent users, users actively hitting the system. Details about how these tests were performed are found in Appendix A, "Performance Tests".

If it is estimated that there will be less than 200 internal Concurrent Users then 1 server should be enough and an additional one if failover is needed. A server should be added for every additional 200 concurrent users.

This is an estimate and will vary depending on how aggressively users are using the system.

Application Server Memory

The memory needs for each application server will depend on the maximum number of logged in users, the size of the user's session, the number of applications running on each server and the base amount of memory needed for each application, see Appendix D for more information. The user's session size will vary from customer to customer. Below is a table with customer observed memory footprints.

Average User Session Size for GSM	GSM user session size is usually determined by customization and will range from 20 MB to 60MB. See Appendix C, "Estimating User Session Size" to accurately measure your average user session size. Otherwise, use one of our three pre-defined user session sizes.
Average User Session Size for NPD	NPD user session size will range from 15 MB to 30 MB. It is typically safe to estimate 20 MB.

For a larger deployment, if we used 40 MB as an average session size, and an average of 3 GB for the total application base footprint, see Appendix D, "Initial Memory Load" for details, then the below table will show the memory requirements.

Total User Session Size	40 MB * 200 users per server = 8GB
Total Application Base - See Appendix D, "Initial Memory Load" for details	3GB
OS Memory	2GB
Total	**13 GB**

Since the amount of memory needed will greatly vary between deployments, it is recommended to measure the actual amount of memory needed, if possible.

DB Server

At this time, assuming the hardware recommendations outlined in the *Agile Product Lifecycle Management for Process Maintenance Pack Installation Guide* are followed, we do not consider the DB to be a performance bottleneck when considering what hardware to buy. Instead, we recommend following the performance and monitoring tips in Chapter 3, "Performance Tips".

File Server

Attachments are stored on the file system with a pointer in the DB. This can be locally on the application server, or on a SAN. We recommend setting an initial disk space allocation for attachments to 25 GB or greater. We further recommend monitoring disk space and adding more when you have reached a predefined threshold (ex. 25% free space remaining).

3

Performance Tips

This chapter includes some best practice steps for optimizing performance of PLM for Process.

Summary

To ensure proper performance of PLM for Process, knowledge of the end to end product architecture is important because performance issues can be introduced at any level. This section outlines a list of best practice steps to help keep your PLM for Process deployment performing well.

Database Server

Maintaining a tuned database is a key component in ensuring the long term health of the overall PLM for Process deployment. Though the applications take advantage of caching when possible, it still relies heavily on database interaction. For this reason, please follow the below recommendations in addition to the "Best Practice" guidelines as recommended by your database vendor.

Fragmentation

SQL Server

When data is inserted into, deleted from, or updated in a SQL Server table, the indexes defined on that table are automatically updated to reflect those changes. As the indexes are modified, the information stored in them becomes fragmented, resulting in the information being scattered across the data files. When this occurs, the logical ordering of the data no longer matches the physical ordering, which can lead to a deterioration of query performance.

To fix this problem, indexes must be periodically reorganized or rebuilt (defragmented) so the physical order of the leaf-level pages matches the logical order of the leaf nodes. This means that you should analyze your indexes periodically to determine whether they've become fragmented and the extent of that fragmentation. From there, you can either reorganize or rebuild the affected indexes, depending on the results of your analysis.

As a helper utility, we have provided a stored procedure in the database called XSP_AGILE_PLM4P_DEFRAG_INDEXES. This stored procedure will analyze the fragmentation of the indexes, generate SQL statements to rebuild or reorganize the over-fragmented ones, and execute those statements. There are 2 values that can be adjusted based on the preference of a knowledgeable DBA. These are the percent average fragmentation and the actual number of fragments per index. These values

help determine which indexes should be defragmented and whether they should be rebuilt or reorganized.

This procedure or another defragmentation routine should be executed on a regular basis to ensure the fragmentation does not affect performance.

Oracle

The Oracle database handles fragmentation internally so defragmenting is not needed.

Optimizer Statistics

Database optimizers rely on statistics to determine the best plan for each query. It is important that these statistics are up to date.

SQL Server

Statistics are gathered automatically in SQL Server so no maintenance is required.

Oracle

The process to maintain updated statistics in Oracle will vary depending on the version. We recommend that you follow the advice of your DBA and consult the Oracle database documentation when determining your plan.

Indexes

PLM for Process comes out of the box with indexes to account for common usage of the application. Since the need for indexes greatly depends on certain characteristics that are specific to each deployment, such as row counts, ongoing analysis will need to be performed to detect the need for additional ones.

It is strongly recommended that the executing SQL is monitored and the proper analysis is performed.

Customer created indexes will not be removed during upgrades and it is recommended to use a naming convention different than the one used for the core indexes.

Summary

Action	SQL Server	Oracle
Defragment	Needed	N/A
Update Optimizer Statistics	N/A	Might be needed
Create new indexes	Needed	Needed

CPU and Memory Consumption

Under sizing the server capacity for the PLM for Process deployment will have a direct impact on performance. Please follow the guidelines in this document to determine the appropriate hardware, taking into account growth.

Monitoring

As user adoption of the system grows, the amount of load put on the system could increase past the initial expectations. If the CPU utilization or application pool

memory consumption increases beyond the optimal range, the user experience will degrade due to slow page response time.

Both the CPU and memory consumption can be monitored by various tools, one being the Windows Performance Monitor. Instructions can be found here: http://technet.microsoft.com/en-us/library/cc749115.aspx

We recommend monitoring the following:

What to monitor	Perfmon Counter	Limit
Database and Application Server CPU Utilization	NeedProcessor / % Processor Timed	Should not exceed 70% for an extended period
Total Application Pool Memory	Process/Private Bytes	Should not exceed 80% of the available memory. Available memory is the total physical memory minus the amount used for the operating system and any other processes.

Caching and Compression

Using caching and compression is the most effective way to reduce page response time. There is a very large performance impact on sites with low bandwidth/ high latency. It is strongly recommended that all sites use the appropriate levels of caching and compression.

Although we provide links to setup caching and compression within IIS, both can be implemented or negated at an upstream device in the architecture. We recommend you discuss options for caching and compression within your IT organization. We also recommend using a utility like IE9 developer tools as a client to verify caching and compression are setup and working correctly.

Caching

Caching content will drastically reduce the amount of round trips between the client browser and the server. By default, IIS7 caches the following extensions.

- .css
- .js
- .jpg
- .gif
- .png
- .ico

Compression

Compression reduces the size of the data passed between the server and the client browser. Because of the compression algorithms are so efficient, we recommend all data be compressed.

For instructions on how to enable compression in IIS 7 please see the following article.

http://technet.microsoft.com/en-us/library/cc754668(v=ws.10).aspx

Troubleshooting Performance Issues

There are many possible causes of performance issues. It is best to eliminate as many variables as possible, in a methodical way, before calling Oracle Support. Below is a list of items to go through.

All tests should be performed after a user executes the exact use case at least one time. The first time through will be slower and is not considered an issue by Oracle.

Task	Steps to take
Isolate performance use cases	Determine the exact pages, actions and data that are involved with the performance issue.
Ensure caching & compression is enabled	Make sure the caching & compression recommendations in this document are followed.
Eliminate custom code	Eliminate all custom code by testing with a vanilla installation. If this improves performance it is likely the issue is in the custom code.
Add database indexes if needed	Monitor and analyze all SQL statements that are executed for requests that are considered slow. Have a DBA determine if new indexes are needed.
Monitor the hardware for capacity limits	Determine if the issue is due to improper sizing of the hardware by performing the test with a single user on the system. If the performance issue cannot be reproduced with a single user then it is likely to be a hardware capacity issue. If the issue only occurs under normal production load, then monitor the application data server for overloading the processor and memory. Check that the IIS application pools are configured according to the recommendations in this document.
Eliminate network hardware	Determine if the issue is due to hardware on the network such as load balancers, reverse proxies or firewalls by testing in an environment that does not have any of these between the application server and the client.
Check network bandwidth and latency	Determine if the issue is due to a poorly performing network.
Monitor log files	Monitor all log files for error conditions that might cause additional overhead. This includes PLM for Process' Event Logs as well as all third party log files such as SSO, LDAP, web logs, and database.
Oracle Support	If you have determined that the issue is within the Oracle code, then open a service request with Oracle Support and provide them as much detail as possible, including the outcome of all the above test. They will need to have the exact steps taken to reproduce the performance issue along with an obfuscated database.

A

Performance Tests

Script Execution

The load testing was performed using scripts that simulated users exercising the GSM and NPD applications. Both applications had two scripts, one performing only reads and one performing writes. They had the following Think Time, time between clicks:

Read User — 20 to 30 seconds

Write User — 40 to 60 seconds

The test started with 1 simulated user and every minute another user was added until the average response time exceeded 5 seconds. The test then runs for a long duration under that load to show stability. The number of users during that test is what was used as the maximum number of users per server.

B

Testing Environment

This appendix contains guidelines for setting up a testing environment.

Load Testing Environment

Application Server	Virtual Machine
	OS: Windows Server 2008 R2
	Web Server: IIS 7.5
	Processor: Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 2995 Mhz, 4 Core (s)
	Total Physical Memory: 32 GB
Database Server	Virtual Machine
	OS: Windows Server 2008 R2
	DB: Oracle 11g R2
	Processor: Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 2995 Mhz, 4 Core(s)
	Total Physical Memory: 16 GB

C

Estimating User Session Size

Session Size Estimation

If it is necessary to get an accurate user session size for your environment, you need to run a load test using load testing software with at least 100 users hitting a single application. For example, take GSM. Prime the application with a single user and measure the baseline application pool memory usage. Execute a load test with 100 users simulating a reasonable think time. At the end of the test, measure the total memory used, subtract the baseline, and divide by the total number of users. This will give you an approximate user session size for GSM.

- User session size = (MB consumed during test – base MB for application pool) / users simulated
- User session size = (6000 MB – 700 MB) / 100 users
- User session size = 53 MB

D

Initial Memory Load

Initial Memory Load

The following is the initial memory footprint observed for each application pool in the recommended application distribution. This represents loading the application with a single user.

Application Pool Name	Applications	Memory (MB)
PLM4P_GSM	GSM	700
PLM4P_GSMView	GSMView	700
PLM4P_PDM	Integration, ProdikaReporting, SCRM, WFA	500
PLM4P_MAIN	Portal, REG, UGM, DRL	700
PLM4P_FC	PQS, OPT	400
PLM4P_NPD	NPD	500
PLM4P_COLLAB	SupplierPortal, SupplierPortalAdmin, eQ	300