



An Oracle White Paper
November 2012
Introduced in Release 6.1.0.1
E38255-01

Capacity Planning

Components and Requirements

Below is a list of the components of the PLM4P application suite and directly related 3rd party hardware devices.

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 a 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 *Agile Product Lifecycle Management for Process Install/Upgrade Guide* located on OTN, <http://www.oracle.com/technetwork/documentation/agile-085940.html#plmprocess>.

Capacity Planning

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 3 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.

App Pool Name	Applications	Reason
PLM4P_GSM	GSM	GSM is a heavily used app and has the potential to consume the most memory. Therefore, we recommend it should always be isolated in its own app pool.
PLM4P_GSMView	GSMView	GSMView is a read-only copy of GSM with the purpose of offloading certain memory intensive operations. It may not be required to be in an isolated app pool, but should be separated from GSM.
PLM4P_PDM	Integration, ProdikaReporting, SCRM, WFA	
PLM4P_MAIN	Portal, REG, UGM, DRL	
PLM4P_FC	PQS, OPT	

App Pool Name	Applications	Reason
PLM4P_NPD	NPD	NPD is a heavily used app and has the potential to consume more memory than other apps. Therefore, we recommend it should be isolated in its own app 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.

See Appendix A for a definition of Active Users.

Size	Number of Active Users	Configuration
Small	Less than 200	Servers 1 Internal Application Production with 4+ cores and 4 GB RAM 1 Internal Data Administration Staging with 2+ cores and 4 GB RAM 1 External Supplier Portal/eQ with 4+ cores and 4 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 6 GB RAM 1 Internal Data Administration Staging with 4+ cores and 4 GB RAM 1 External Supplier Portal/eQ with 4+ cores and 4GB RAM 1 Internal DB Additional Hardware Load Balancer (recommended) Reverse Proxy (recommended)

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

Sizing & Scaling

Application Server

For sizing information on the Application Server, use the PLM4P_Server_Calculator.xlsx spreadsheet. This will allow you to estimate the number of servers, cores per server, and memory per server required based on your concurrent users.

Note: If you are going to use the calculator, please know that these are rough recommendations. Often times, the recommended memory per server will be less than 8GB. However, memory is a low cost upgrade and purchasing over the recommendation should be considered when purchasing hardware. This will help to future-proof your hardware.

To obtain this spreadsheet, please contact support or your sales representative.

There are 2 worksheets to be aware of; the Concurrent Users worksheet and the Application Server worksheet.

Concurrent Users Worksheet

Start with estimating your number of registered users for GSM and NPD. For the purposes of this calculator, your user base should represent distinct users and not overlap. Given the following inputs, your output will be a rough estimate of your concurrent users for the system. This will then be carried over to the Application Server worksheet.

User Communities Estimated Registered Users *	This represents the expected registered users for GSM and NPD some of the more common user communities.
Global Distribution Modifier (GDM)	The GDM represents the significant regions based on time zone. If the application will be used in all corners of the world use a value of 0.5. It is recommended to use values of 0.5, 0.75, or 1.0 based on your distribution
% of Registered Users Active	Active Users represents the percentage of registered users you would expect on a daily basis. If 100% of the users are expected to log into the application, manage specification, participate in projects, or manages suppliers requires a value of "1.0".
Concurrency Factor	Concurrency Factor represents the percentage of active users that would be expected to concurrently interact with the system at a single point in time. Usually set between 0.1 and 0.2.

* For accurate calculations, the users should not overlap.

Application Server Worksheet

Given the following inputs, your output will be a rough recommendation of the application server hardware needs to support your users. You should come away with an understanding of the total number of application servers, the memory, and the number of cores for each server.

GSM Concurrent Users **	GSM concurrent users represent a portion of the overall total users that will simultaneously be working in GSM. This number typically represents 10% of your active user base for GSM.
NPD Concurrent Users **	NPD concurrent users represent a portion of the overall total users that will simultaneously be working in NPD. This number typically represents 10% of your active user base for NPD.
All Other Concurrent Users	All other concurrent users represent a portion of the overall total users that will simultaneously be working in all apps not represented by NPD or GSM. This number typically represents 10% of your active user base for all other apps.

Average User Session Size for GSM	GSM user session size is usually determined by customization and will range from 8 MB to 15 MB. 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 6 MB to 8 MB. It is typically safe to estimate 7 MB.
OS Memory	OS Memory will vary based on the customer’s environment. We have estimated 1000 MB for base operations.
Growth Factor	Growth of customer’s user base over a 3 year period.

** By default, this cell will pull the value from the Concurrent Users worksheet

DB Server

At this time, assuming the hardware recommendations outlined in the Install 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 the Performance Tips section of this document.

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).

Load Testing Results

Below you will find summary results for our load tests. At incremental points in the graph, we measured the response time, CPU Utilization, and App Pool Memory for the number of concurrent users given. For a given graph, the row highlighted in green represents the last good measurement before we reached our breakpoint of 5 seconds.

Although we feel our data and use cases represent a typical customer, they will differ from yours. For this reason, we always recommend performing your own load and performance testing.

Global Specification Management (GSM) – 1 core

Number of Concurrent Users	Average Response Time (in seconds)	CPU Usage		App Pool Memory
		Application Server	Database	
50	1.7	45%	13%	586 MB
90	4.9	79%	22%	909 MB
100	8.4	83%	23%	989 MB
120	15.9	83%	24%	1086 MB

Global Specification Management (GSM) – 2 cores

Number of Concurrent Users	Average Response Time (in seconds)	CPU Usage		App Pool Memory
		Application Server	Database	
50	1.7	23%	15%	626 MB
100	2.3	58%	28%	1009 MB
120	3.64	73%	31%	1156 MB

Global Specification Management (GSM) – 4 cores

Number of Concurrent Users	Average Response Time (in seconds)	CPU Usage		App Pool Memory
		Application Server	Database	
50	1.6	13%	14%	672 MB
100	1.6	30%	28%	1068 MB
150	2.2	51%	41%	1384 MB
170	3.5	62%	45%	1585 MB
200	11.0	65%	45%	1907 MB

New Product Development (NPD) – 2 cores

Number of Concurrent Users	Average Response Time (in seconds)	CPU Usage		App Pool Memory	
		Application Server	Database		
50	1.1	23%	10%	446 MB	
100	1.6	57%	17%	725 MB	
137	4.4	83%	21%	894 MB	
150	10.0	89%	23%	1002 MB	

GSM and NPD – 4 cores

Number of Concurrent Users	Average Response Time (in seconds)	CPU Usage		App Pool Memory	
		Application Server	Database	GSM	NPD
50	1.34	12%	11%	412 MB	513 MB
100	1.33	28%	23%	584 MB	673 MB
150	1.36	42%	32%	735 MB	782 MB
200	1.56	61%	42%	951 MB	951 MB
246	4.57	80%	49%	1225 MB	1090 MB
250	5.50	83%	50%	1260 MB	1099 MB
300	11.15	90%	53%	1435 MB	1196 MB

Performance Tips

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 two 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
DB & App Server CPU Utilization	Processor / % Processor Time	Should not exceed 70% for an extended period
Application Pool Memory	Process/Private Bytes	< 800 Mb on 32 bit <1800Mb on 64 bit

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](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.

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	<p>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.</p> <p>If the issue only occurs under normal production load, then monitor the application data server for overloading the processor and memory.</p> <p>Check that the IIS application pools are configured according to the recommendations in this document.</p>
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 tests. They will need to have the exact steps taken to reproduce the performance issue along with an obfuscated database.

Appendix A - Definitions

Definitions

Registered User	Person who uses the PLM4P application and provides his or her credentials to login.
Active User	Person who uses the PLM4P application, provides his or her credentials to login, and is currently logged in and working. This is a percentage of registered users you would expect to be logged in on a daily basis.
Concurrent User	Person who is logged in and is accessing an application resource at the same time as one or more other users.
User Session Size	Memory allocated to a single user.

Appendix B – Testing Environment

Load Testing Environment

Application Server	<p>Virtual Machine</p> <p>OS: Windows Server 2008 R2</p> <p>Web Server: IIS 7.5</p> <p>Processor: Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 2995 Mhz, 4 Core(s)</p> <p>Total Physical Memory: 5.86 GB</p>
Database Server	<p>Virtual Machine</p> <p>OS: Windows Server 2008 R2</p> <p>DB: MSSQL 2008</p> <p>Processor: Intel(R) Xeon(R) CPU X5670 @ 2.93GHz, 2995 Mhz, 4 Core(s)</p> <p>Total Physical Memory: 5.86 GB</p>

Appendix C – Estimating User Session Size

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 app. For example, take GSM. Prime the app 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 app pool) / users simulated

User session size = (1500 MB – 300 MB) / 100 users

User session size = 12 MB

Appendix D – 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.

App Pool Name	Applications	Memory (MB)
PLM4P_GSM	GSM	150
PLM4P_GSMView	GSMView	150
PLM4P_PDM	Integration, ProdikaReporting, SCRM, WFA	220
PLM4P_MAIN	Portal, REG, UGM, DRL	340
PLM4P_FC	PQS, OPT	180
PLM4P_NPD	NPD	150
PLM4P_COLLAB	SupplierPortal, SupplierPortalAdmin, eQ	190



Capacity Planning
November 2012

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2012, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 1010

Hardware and Software, Engineered to Work Together