Legal Notices

Trademark Notice

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

License Restrictions, Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

.

Introduction

Oracle Insurance Policy Administration (OIPA) System v11 is the new release of Oracle's suite of products for the Insurance Industry that has the central role in directing the life cycle of a policy as defined for various products. The OIPA system has a crucial role in automating the policy administration, using defined business work flows in the product definitions. The life cycle progression and automated rule execution is achieved in OIPA by means of the Cycle sub-system.

OIPA Cycle is a powerful offline, automated sub-system built into OIPA to analyse each policy, identify the life cycle actions to be triggered and execute such actions. For example, such actions could be trigger a premium due task (and its associated sub-tasks if so defined), or execute a fund balancing rule for ULIP. Oracle Insurance Policy Administration uses a carrier configured set of products (plans) and executes business events to move policies through their natural life cycle. It sets up policies in a set of states that have been identified as representative set of a large enterprise carrier implementing the full range of products OIPA is capable of handling.

OIPA Web is the user interface for carriers to provide for their employees / analysts and other operators to the policy system. This channel is expected to service a lower volume of activity than Cycle, however, both are designed to scale to any size necessary.

The new features introduced in the release are:

- New UI using Oracle jet technology.
- Configurable widgets for various categories.

This document presents the benchmarking results of OIPA v11. The results are used by Oracle to further analyze and prepare recommendations for standard OIPA configuration and how these can be determined, presented in the below sections.

Application Server Settings

PAS Properties

S.No	Settings	Description
1	application.resourceCacheTimeout=-1	The cache time out should be -1 to never check for updates for translations
2	application.mode=PRODUCTION	application mode to be set to PRODUCTION for performance testing
3	debug.remoteDebugging=No	The remote debugging should be turned off
4	jpa.showSql=false	showSql should be 'false' to have no sql statements return to server logs
5	» application.databaseType=SqlServer » application.databaseType=Oracle » application.databaseType=DB2	set the database type based on the database used for testing
6	search.field.text.caseInsensitive=false	set **case-insensitve** search to false for search case sensitive searches for Oracle database. For DB2, Setting it to "true" would enable the code to use the generated columns for comparison. refer the JIRA(OIPA-136,

		OIGPA-157) as well to understand the logic.
7	#jpa.data-basePlatform=org.eclipse.persistence.platform.database.OraclePlatform #jpa.databasePlatform=org.eclipse.persistence.platform.database.DB2Platform #jpa.data-basePlatform=org.eclipse.persistence.platform.database.SQLServerPlatform #jpa.databasePlatform=com.adminserver.dal.jpa.CustomDB2Platform	The jpa database platform to be used with top link essen-tials. For DB2, use Cus-tomDB2PlatForm JPA for top link essentials. **Note:** This fix is not required if using EclipseLink (OIPA-136, OIGPA-157)

Weblogic Settings

S.No	Settings	Description
1	Login Timeout =25000	<ManagedServer> -- Configuration -- tuning -- Login Timeout should be 25000
2	HTTP Duration = 60	<ManagedServer> -- Configuration -- Protocols -- HTTP -- Duration should be 60
3	Initial Capacity=250 Increment = 10	Services --<DataSource> --Configuration -- Con-nectionPool -- Initial and Maximum Capacity should be 250 and Incriment should be 10
4	GC Algorithm "-XX:+UseG1GC"	G1GC to be part of jvm arguments
5	-Xms4g –Xmx4g	Heap memory settings

HTTP Server Settings

Note: These are recommended settings could change based on testing effort and any issues found

S.No	Settings	Description
1	Timeout = 1200	The number of seconds before receives and sends time

		out.
2	KeepAlive = On	Whether or not to allow persistent connections (more than #1 request per connection). Set to "Off" to deactivate.
3	MaxKeepAliveRequests = 600	The maximum number of requests to allow during a persistent connection. Set to '0' will allow an unlimited amount. **Note:** We recommend you leave this number high, for maximum performance.
4	KeepAliveTimeout =100	Number of seconds to wait for the next request from the same client on the same connection.
5	ThreadLimit = 25	Maximum setting of ThreadsPerChild
6	ServerLimit = 64	Maximum setting of StartServers
7	StartServers = 1	Initial number of server processes to start
8	MaxClients = 800	Maximum number of simultaneous client connections
	MinSpareThreads = 25	Minimum number of worker threads which are kept spare
	MaxSpareThreads = 50	Maximum number of worker threads which are kept spare
	ThreadsPerChild = 25	Constant number of worker threads in each server process
	MaxRequestsPerChild = 0	Maximum number of requests a server process serves

Application Server Settings

PAS Properties

S.No	Settings	Description
1	application.resourceCacheTimeout=-1	The cache time out should be -1 to never check for updates for translations
2	application.mode=PRODUCTION	application mode to be set to PRODUCTION for performance testing
3	debug.remoteDebugging=No	The remote debugging should be turned off
4	jpa.showSql=false	showSql should be 'false' to have no sql statements return to server logs
5	» application.databaseType=SqlServer » application.databaseType=Oracle » application.databaseType=DB2	set the database type based on the database used for testing
6	search.field.text.caseInsensitive=false	set **case-insensitve** search to false for search case sensitive searches for Oracle database. For DB2, Setting it to "true" would enable the code to use the generated columns for comparison. refer the JIRA(OIPA-136,

		OIGPA-157) as well to understand the logic.
7	#jpa.data-basePlatform=org.eclipse.persistence.platform.database.OraclePlatform #jpa.databasePlatform=org.eclipse.persistence.platform.database.DB2Platform #jpa.data-basePlatform=org.eclipse.persistence.platform.database.SQLServerPlatform #jpa.databasePlatform=com.adminserver.dal.jpa.CustomDB2Platform	The jpa database platform to be used with top link essen-tials. For DB2, use Cus-tomDB2PlatForm JPA for top link essentials. **Note:** This fix is not required if using EclipseLink (OIPA-136, OIGPA-157)

Weblogic Settings

S.No	Settings	Description
1	Login Timeout =25000	<ManagedServer> -- Configuration -- tuning -- Login Timeout should be 25000
2	HTTP Duration = 60	<ManagedServer> -- Configuration -- Protocols -- HTTP -- Duration should be 60
3	Initial Capacity=250 Increment = 10	Services --<DataSource> --Configuration -- Con-nectionPool -- Initial and Maximum Capacity should be 250 and Incriment should be 10
4	GC Algorithm "-XX:+UseG1GC"	G1GC to be part of jvm arguments
5	-Xms4g –Xmx4g	Heap memory settings

HTTP Server Settings

Note: These are recommended settings could change based on testing effort and any issues found

S.No	Settings	Description
1	Timeout = 1200	The number of seconds before receives and sends time

		out.
2	KeepAlive = On	Whether or not to allow persistent connections (more than #1 request per connection). Set to "Off" to deactivate.
3	MaxKeepAliveRequests = 600	The maximum number of requests to allow during a persistent connection. Set to '0' will allow an unlimited amount. **Note:** We recommend you leave this number high, for maximum performance.
4	KeepAliveTimeout =100	Number of seconds to wait for the next request from the same client on the same connection.
5	ThreadLimit = 25	Maximum setting of ThreadsPerChild
6	ServerLimit = 64	Maximum setting of StartServers
7	StartServers = 1	Initial number of server processes to start
8	MaxClients = 800	Maximum number of simultaneous client connections
	MinSpareThreads = 25	Minimum number of worker threads which are kept spare
	MaxSpareThreads = 50	Maximum number of worker threads which are kept spare
	ThreadsPerChild = 25	Constant number of worker threads in each server process
	MaxRequestsPerChild = 0	Maximum number of requests a server process serves

Cycle Settings

PAS Properties

S.No	Settings	Description
1	N/A	Agent should be deployed on the web container with appropriate cycle agent war
2	application.mode=PRODUCTION	application mode should be set to PRODUCTION where ever applicable

Cycle Client Settings

S.No	Settings	Description
1	cycle.period=5	set the cycle period for cycle process to sleep before waking up
2	cycle.batchSize=2000	The maximum number of cycle tasks that can be logged with the thread executor. This number, plus the thread count, gives us the maximum number of tasks that can be checked out for processing.
3	cycle.groupSize=100	The maximum number of tasks to be grouped together for execution.
4	application.resourceCacheTimeout=-1	Time to cache translations, in minutes, before checking data source for updates. Which should be set to 'never'
5	application.mode=PRODUCTION	application mode is to be set to PRODUCTION
6	debug.remoteDebugging=No	remote debugging should be turned off
7	jpa.showSql=false	showSql should be false

Database Settings

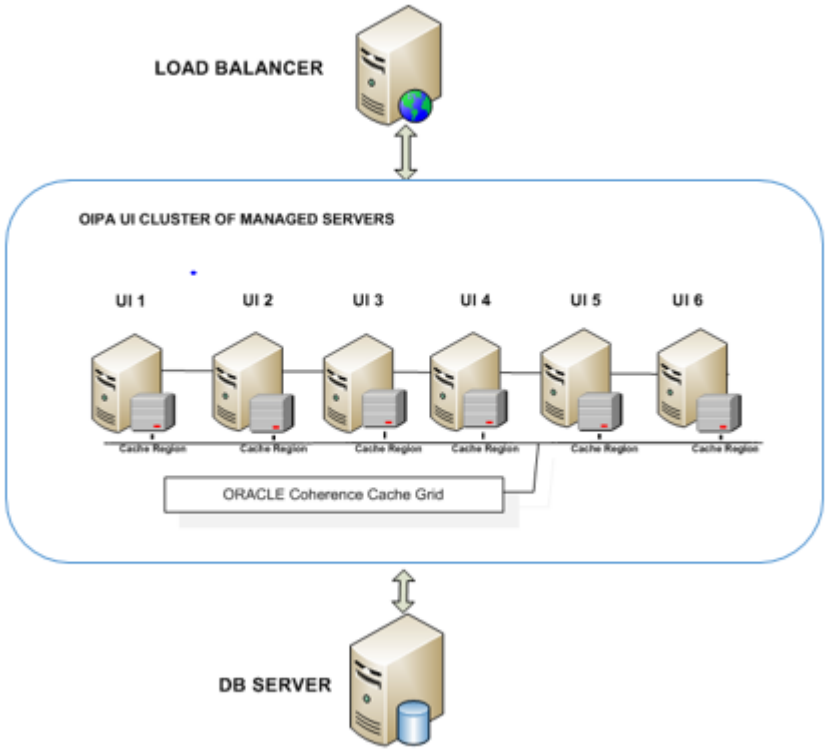
S.No	Settings	Description
1	N/A	Run Strip Utility immediately after importing a new database.
2	N/A	Import/ Export should prefix today's date to log to import or export files.
3	DBMS_STATS.GATHER_TABLE_STATS ('ASADMIN', 'ASCYCLE', cascade=>true, degree => 32);	The table statistics can be included in all the cycle related procedures which populate the CYCLE tables. The below has been added to the 'ASC_POPULATEPOLICYCCYLE' procedure.
4	select 'ALTER SEQUENCE '||sequence_name||' CACHE 10000;' from all_sequences where sequence_owner='ASADMIN';	SEQUENCE Cache, Consider setting CACHE to 10000 for all oracle sequences. Use the given sql statement to generate the ddl for altering sequence cache.
5	alter system set cursor_sharing='FORCE' scope=spfile;	Cursors do not seem to be reused well. Change cursor_sharing to FORCE. The cursor_sharing was set to 'SIMILAR' which is deprecated in Oracle 11.2G
6	alter system set Processes='4000' scope=spfile;	Set the processes **Note:** These parameters might depend on the load execution and weblogic connection settings
7	alter system set open_cursors='30000' scope=spfile;	Set the open_cursors **Note:** These parameters might depend on the load execution
8	alter system set session_cached_cursors='3000' scope=spfile;	Set session_cached_cursors **Note:** These parameters might depend on the load execution
9	alter system set memory_target='8G' scope=spfile;	set memory target **Note:** These parameters might depend on the load execution
10	alter system set shared_pool_size='1G' scope=spfile;	Set shared_pool_size **Note:** These parameters might depend on the load execution
11	» Redo Log file configuration should be Groups -3	set the redo log file configuration

	» Members in each group -2 » Size of Each Member - 2GB Place the 2 members of each group in separate file systems built on different disk arrays	

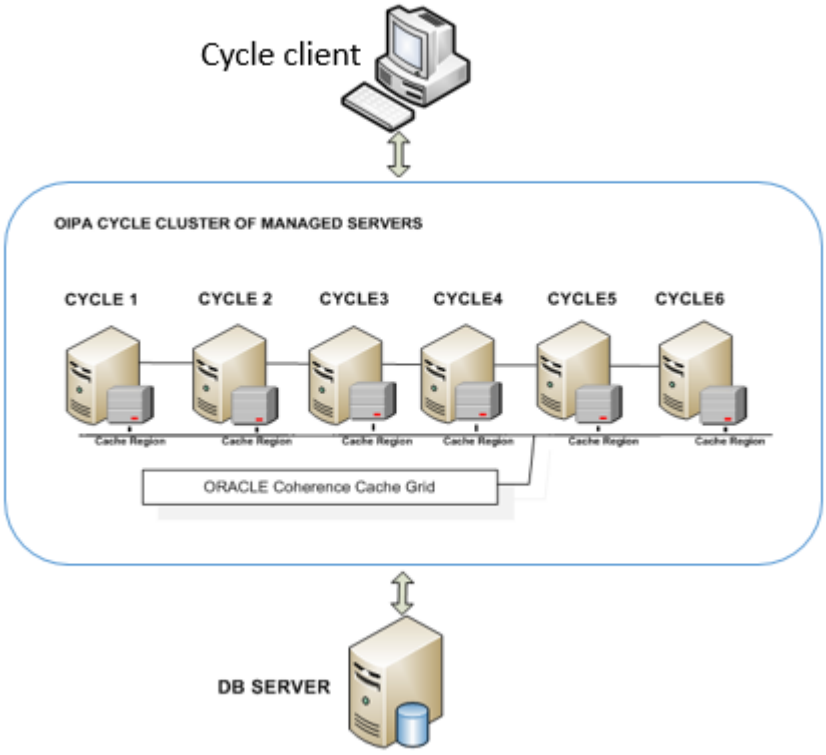
Test Environment Set-up

The following environment set-up is used to test the OIPA application. It is configured as per the Oracle OIPA standard hardware, database, and application server specifications using the standard OIPA performance test data sets.

OIPA UI



Cycle


Cycle client
OIPA CYCLE CLUSTER OF MANAGED SERVERS
CYCLE 1
CYCLE 2
CYCLE3
CYCLE4
CYCLE5
CYCLE6
ORACLE Coherence Cache Grid
DB SERVER

Hardware Specification

Application Server

Processor	6-core Intel(R) Xeon(R) CPU X5670 @ 2.93GHz {2 logical cores per physical core, supports 12 threads}
RAM	148 GB
Disk Space	1.9 TB
Operating System	Oracle Linux Server release 6.7
Application Server	Oracle WebLogic 12c (12.2.1.1.0)
Java version	"1.8.0_77" Hotspot(TM) 64-bit

Database Server

Processor	8-core SPARC T4 processor @ 2.85 GHz { 8 logical cores per physical core, Supports 64 threads}
RAM	128 GB
Disk Space	2 TB
Operating System	Oracle Solaris 11.1
Database	Oracle 12 Enterprise Edition (12.1.0.1.0)

Load Balancer Server

Processor	2-core Intel(R) Xeon(R) CPU X5675 @3.07 GHz { 2 logical cores per physical core, supports 4 threads}
RAM	15.5 GB
Operating System	Oracle Linux Server release 6.7
HTTP Server	Oracle HTTP Server 12 (is used as the software load balancer)

Configuration used for Cycle and UI Load Tests

This section describes the Weblogic Server Setup and the Cycle property details. For the property descriptions, refer the Glossary section.

Weblogic Server Setup

The Weblogic Server Setup is configured as following:

JDK	JDK 1.8.0_77
Weblogic Servers	2 Linux servers are utilized to setup the WebLogic Cluster.
OIPA & UI Cycle	There are a total of 6 OIPA UI and Cycle instances running on 2 Weblogic servers. Each Linux server has 3 OIPA UI and Cycle instances started.
JVM Memory	The minimum and maximum memory for each JVM is set to 4 GB.
DB Pool Size	The data source connection pool size initially is set to 15, with a maximum of 200

Cycle.Client Properties

The client properties are configured with the following values:

Property	Value
Batch Size	5,000
Grid Task Submission Thread Pool Size	25
Cycle Group Size	75
Cycle Period	5

Cycle.Web Properties

The web properties are configured with the following values:

Property	Value
Grid Task Submission Thread Pool Size	50
Application Mode	PRODUCTION
Application Resource Cache Timeout	-1

Cycle.Web Coherence Properties

The web coherence properties are configured with the following values:

Property	Value

Default Task Processor Thread pool size	90

UI PAS.Properties

The UI properties are configured with the following values:

Property	Value
Application Mode	PRODUCTION
Application Resource Cache Timeout	-1
jpa.showSql	false

Database Settings

The database is configured with the following settings:

- Reorganized tables with large amounts of data
- System Global Area (SGA) Setting was updated from 8G to 18G
- Gathered schema level statistics before each cycle execution

Property	Value
MEMORY_MAX_TARGET	51G
MEMORY_TARGET	51G
system session-cached-cursors	7000
COMMIT_LOGGING	IMMEDIATE
COMMIT_WAIT	NOWAIT
CURSOR_SHARING	FORCE
number of processes	30000
open cursors	30000

Performance Monitoring Tools

jVisualVM

JDK out of box monitoring tool to understand the CPU, Memory, Thread and loaded classes.

Thread Logic

Thread Logic is a GUI based thread dump parsing tool, which performs the following major tasks:

Memory Analyzer Tool

The Eclipse Memory Analyzer is a fast and feature-rich Java heap analyzer tool that helps in finding the memory leaks and reduce memory consumption. The Memory Analyzer is used to analyze productive heap dumps with hundreds of millions of objects, quickly calculate the retained sizes of objects, see which is preventing the Garbage Collector from collecting objects, and run a report to automatically extract leak suspects.

Oracle DB Reports

It is an Automatic Workload Repository (AWR) used to collect performance statistics on,

- Wait events used to identify performance problems.
- Time model statistics indicating the amount of DB time associated with a process from the V$SESS_TIME_MODEL and V$SYS_TIME_MODEL views.
- Active Session History (ASH) statistics from the V$ACTIVE_SESSION_HISTORY view.
- Some system and session statistics from the V$SYSSTAT and V$SESSTAT views.
- Object usage statistics.
- Resource intensive SQL statements.

Oracle 10g introduced the Active Session History (ASH) as part of the Diagnostics and Tuning Pack. It samples information from the [G]V$ views allowing to see current and historical information about active sessions on the database.

The Automatic Database Diagnostic Monitor (ADDM) analyzes data in the Automatic Workload Repository (AWR) to identify potential performance bottlenecks. For each of the identified issues, it locates the root cause and provides recommendations for correcting the problem. An ADDM analysis task is performed and its findings and recommendations stored in the database every time an AWR snapshot is taken provided the STATISTICS_LEVEL parameter is set to TYPICAL or ALL. The ADDM analysis includes the following:

- CPU load
- Memory usage
- I/O usage
- Resource intensive SQL
- Resource intensive PL/SQL and Java
- RAC issues
- Application issues
- Database configuration issues
- Concurrency issues
- Object contention

nmon

Nmon is short for Nigel's performance Monitor for Linux on POWER, x86, x86_64, Mainframe & now ARM (Raspberry Pi). It is a systems administrator, tuner, benchmark tool that gives a huge amount of important performance information in one go. It can output the data in two ways:

1. Display the data on screen (console, telnet, VNC, putty or X Windows) using curves for low CPU impact, which is updated once every two seconds. The user can hit single characters on the keyboard to enable/disable various sorts of data.
2. Display the CPU, memory, network, disks (mini graphs or numbers), file systems, NFS, top processes, resources (Linux version & processors) and on Power micro-partition information.

Results

Cycle Results

Multiple Cycle load tests were performed and documented the best results fetched out of optimal settings. Total 8 rounds of successful Cycle tests had run by altering and adjusting the various JVM and DB parameters.

Thread Pool Size	Total Time	Activities/min	Policies/min
90	68 min	3111.80	246.44

UI Test Results

Multiple rounds of UI load tests performed by applying various changes such as GZIP compression, verifying of various GC algorithms and concluded with G1 GC policy. Generic DB tuning done and applied recommended indexes.

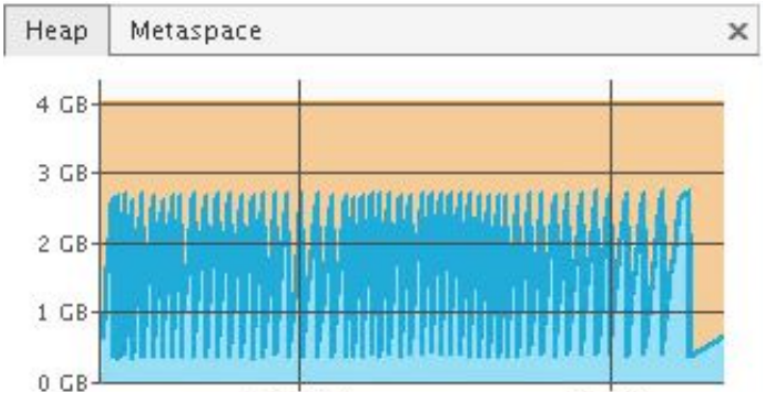
Number of Users	Duration	Average Response Times
200 users	1 hour	0.536 seconds

System Resource Utilization Statistics for Cycle

The system resource utilization (JVM utilization) statistics for all the servers obtained for Cycle load tests are given below:

Heap Utilization

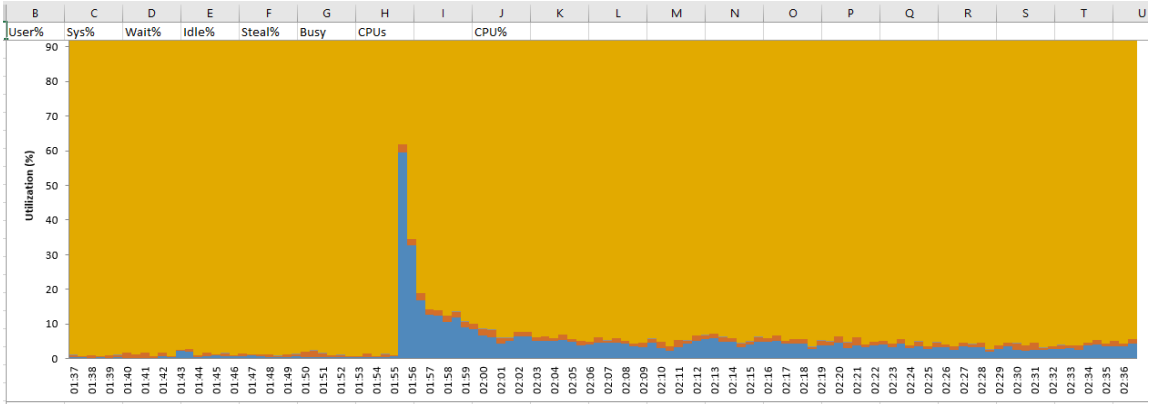
The G1 GC policy enabled optimal memory utilization with appropriate GC activity. The heap usage for each instance is between 1 GB and 3 GB out of 4GB allocation. Frequent minor and major GC's were happened as expected to reclaim young gen memory and decent OLD gen GC's. There is no instance of Out of Memory issues at all.



Application Servers

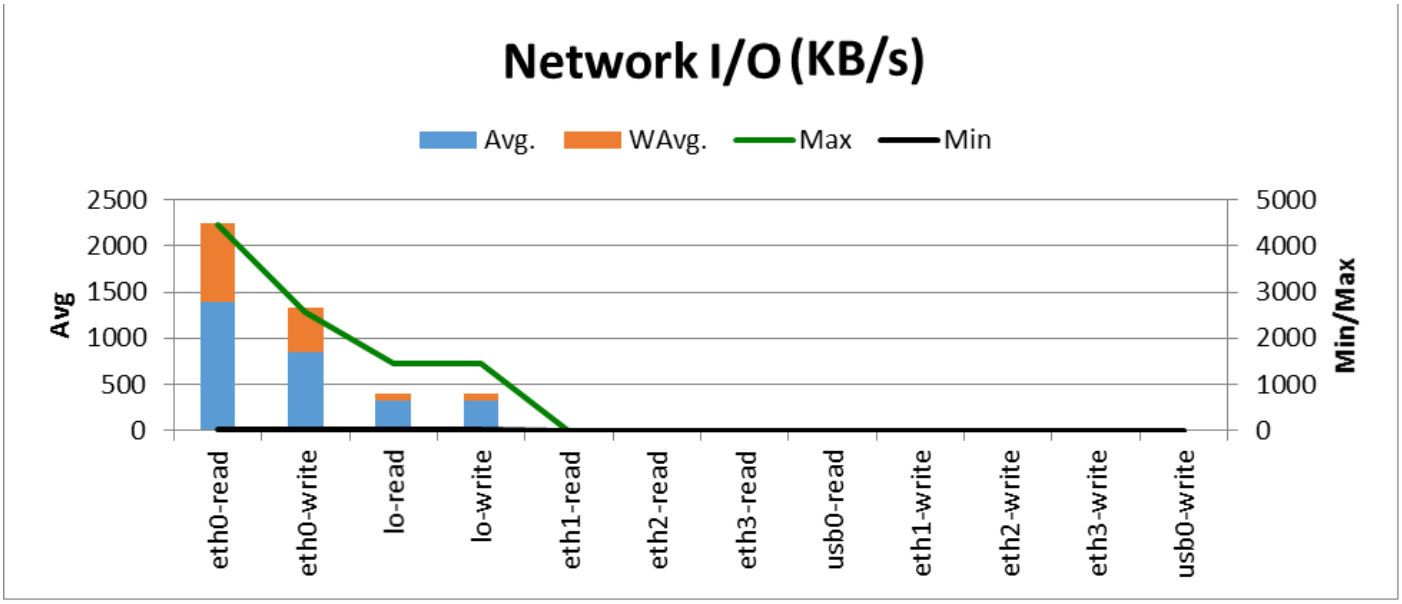
Application Server1 CPU Utilization - Average – 5.14 %, Max – 60 %

Below is the graph that shows the CPU utilization of Application Server1 i.e. 5.14 % average utilization has been recorded and went up to 60 % of maximum utilization at the start of the test.



App Server1 Network Utilization

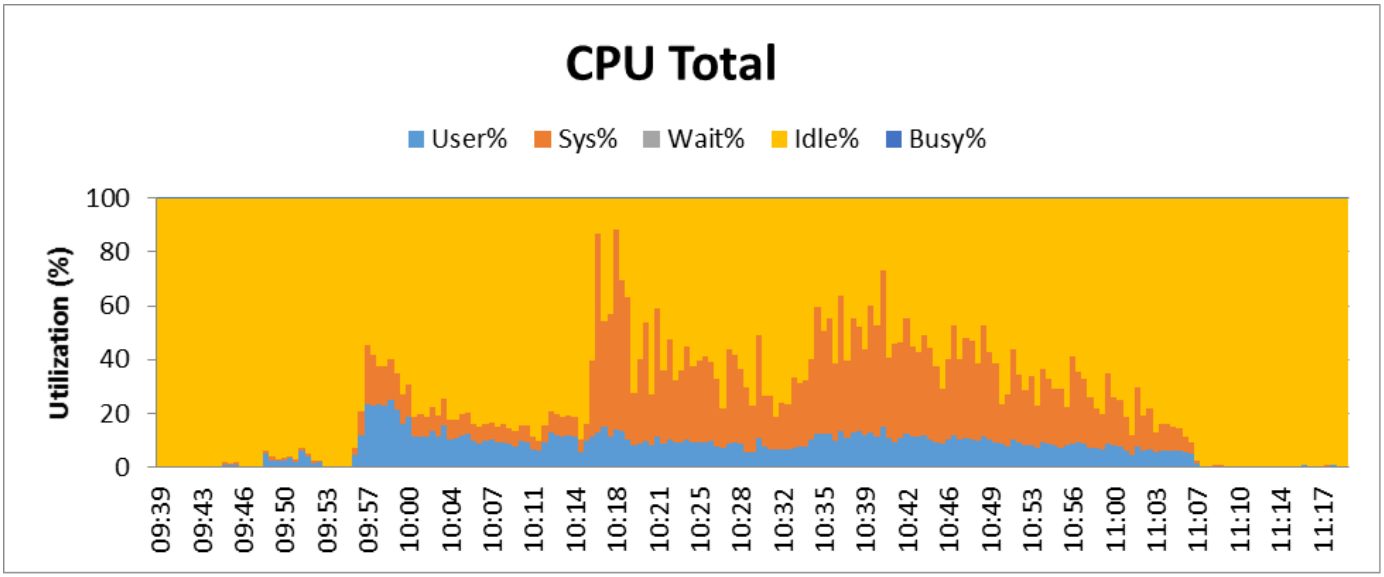
Below is the graph that shows Network utilization of Application Srever1. It can be observed that a series of network reads and writes during the application transaction processing. The average reads are below 1399.3 kb/s and the average writes are below 852.3 kb/s. This indicates the network I/O is decent.



Database Servers

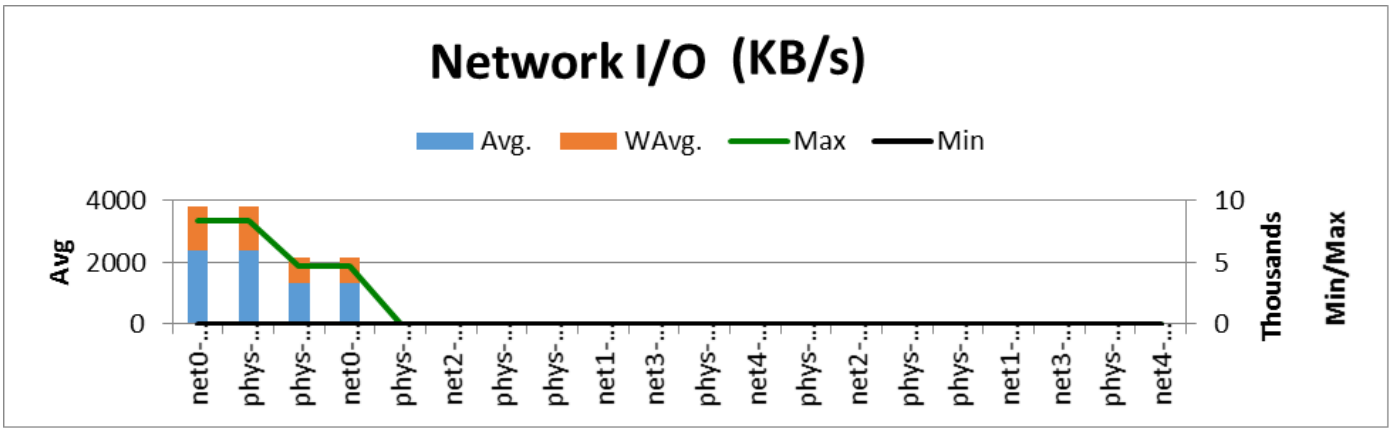
DB Server CPU Utilization - Average – 23.68 %

Below is the graph that shows CPU utilization of a DB server. 23.68% average utilization has been recorded and went up to 90% of maximum utilization.



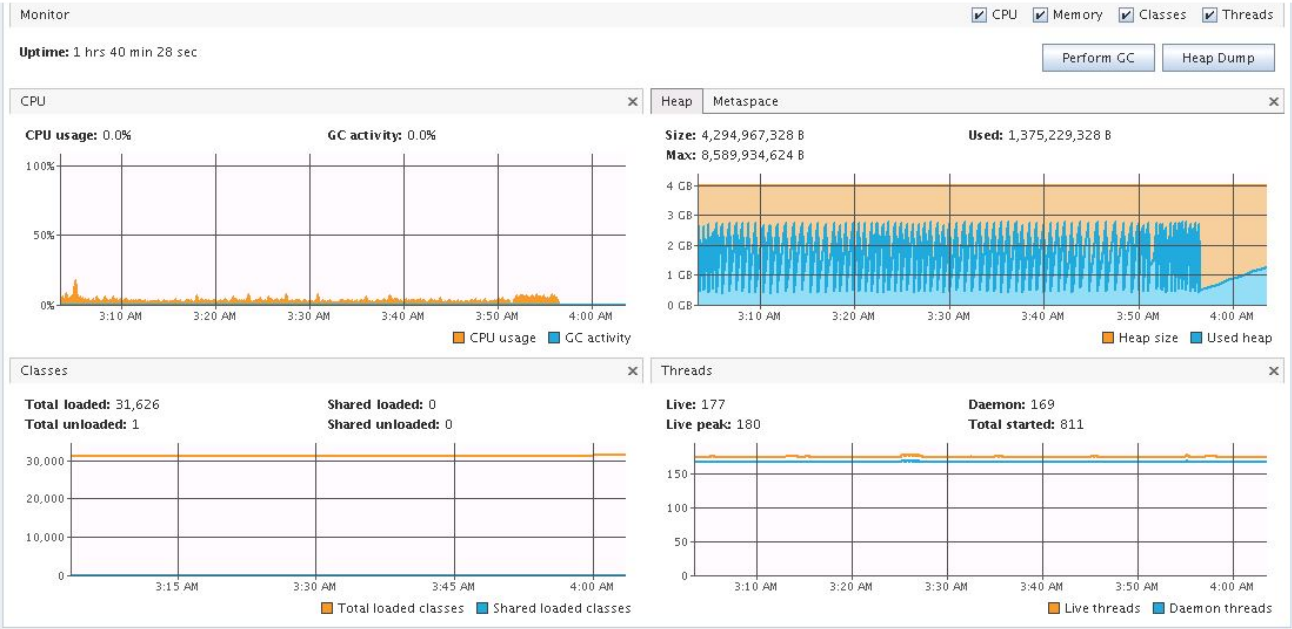
DB Server Network Utilization

Below is the graph that shows network utilization of a DB server. It can be observed that a series of network reads and writes during the application transaction processing. The average reads are below 1337.5 kb/s and the average writes are below 2374 kb/s. This indicates the network I/O is decent.



Application Tier

Both the JVM memory and CPU on the application server are utilized efficiently. Tuning of application properties, jvm parameters and DB connection pool settings as specified above, gave optimal performance.



Database Tier

Applying indexes, reorganizing of tables with large amounts of data and gathering the schema level statistics before each cycle execution improved the database performance.

```
create index OIPAPERF.IDX$$_00010001 on OIPAPERF.ASFINANCIALENTRY
("ACTIVITYGUID","FINANCIALENTRYGUID","ACTIVETODATE");
create index OIPAPERF.IDX$$_00010031 on OIPAPERF.ASMAPCRITERIA
("MAPCRITERIANAME","TEXTVALUE");
create index OIPAPERF.IDX$$_00010031 on OIPAPERF.ASMAPCRITERIA
("MAPCRITERIANAME","TEXTVALUE");
dbms_sqltune.accept_sql_profile(task_name => 'SYS_AUTO_SQL_TUNING_TASK', object_id
=> 10190, task_owner => 'SYS', replace => TRUE);
```

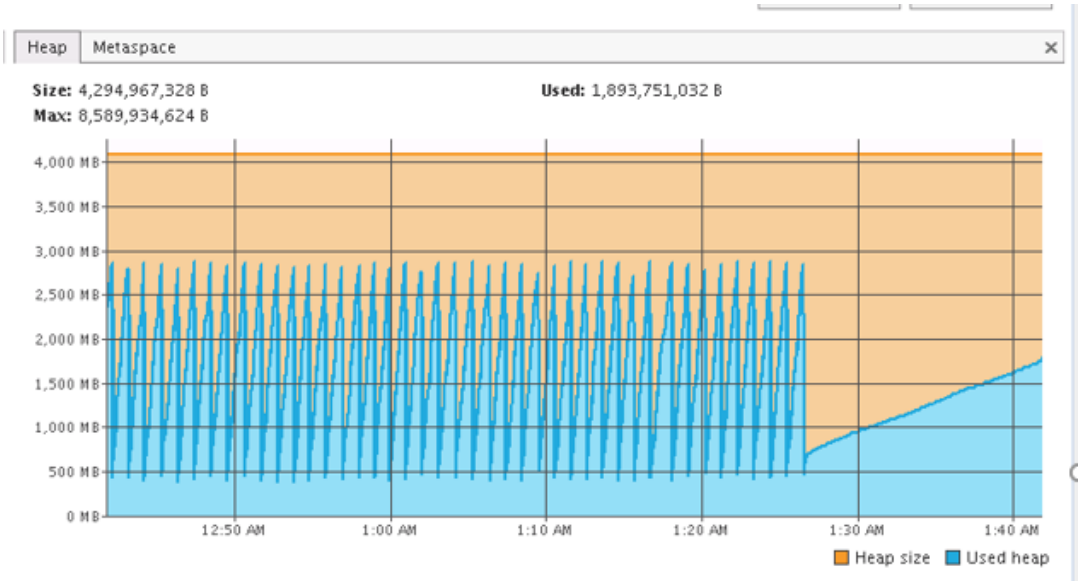
```
dbms_sqltune.accept_sql_profile(task_name => 'SYS_AUTO_SQL_TUNING_TASK', object_id
=> 10127, task_owner => 'SYS', replace => TRUE);
dbms_sqltune.accept_sql_profile(task_name => 'SYS_AUTO_SQL_TUNING_TASK', object_id
=> 10224, task_owner => 'SYS', replace => TRUE);
```

System Resource Utilization Statistics for UI

The system resource utilization (JVM utilization) statistics for all the servers obtained for OIPA UI load tests are given below:

Heap Utilization

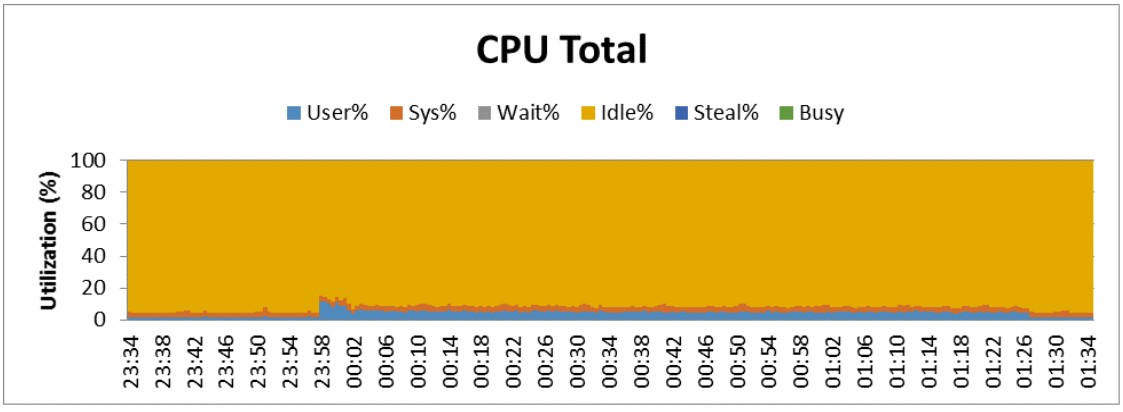
The heap usage for each instance is between 1 GB and 3 GB out of 4GB allocation. Frequent minor GC's are happening as expected to reclaim young gen memory and decent major GC's are also happening to reclaim old gen memory. There is no instance of Out of Memory issues at all.



Application Servers

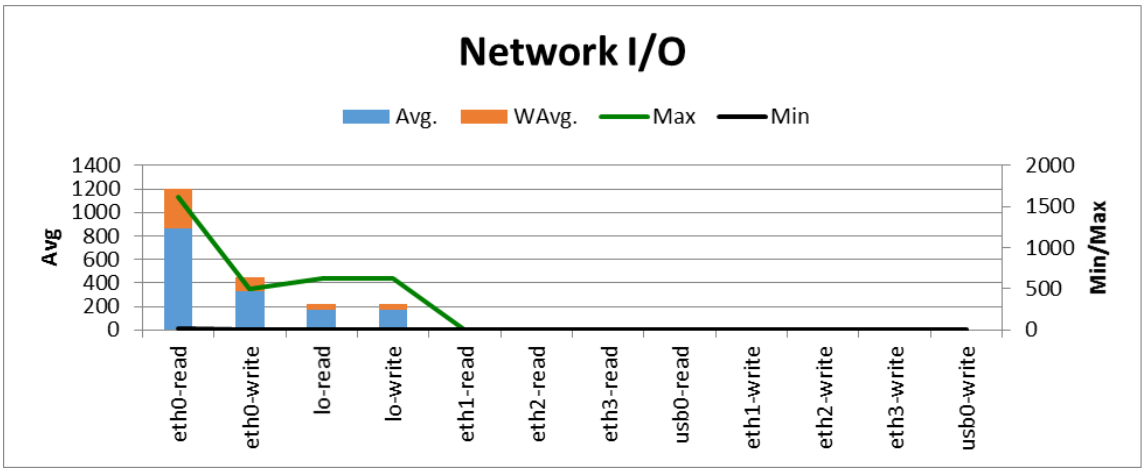
Application Server1 CPU Utilization - Average – 7.65%, Max – 10.2%

Below is the graph that shows the CPU utilization of Application Server1. 7.65 % average utilization has been recorded and went up to 10.2% of maximum utilization.



App Server1 Network Utilization

Below is the graph that shows Network utilization of Application Srever1. It can be observed that a series of network reads and writes during the application transaction processing. The average reads are below 862.2 kb/s and the average writes are below 328.1 kb/s. This indicates the network I/O is decent.



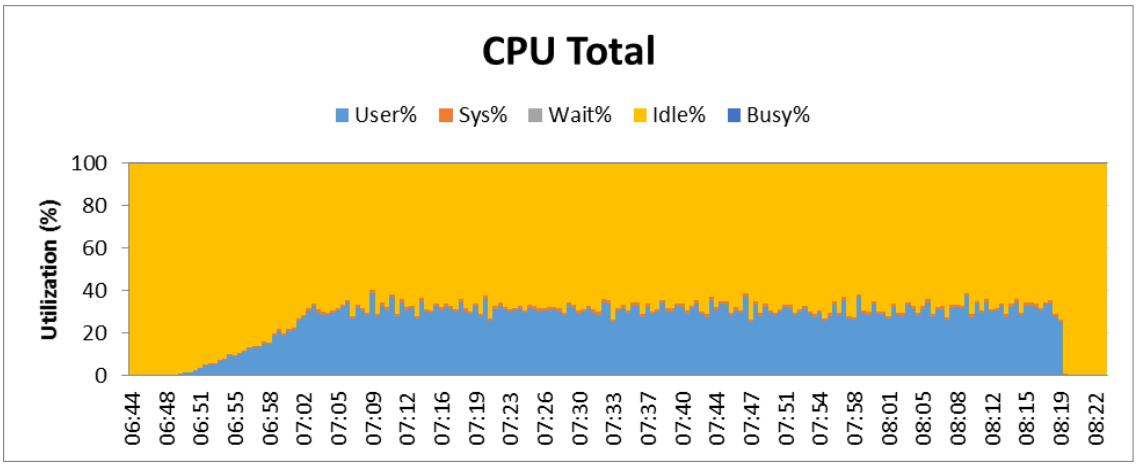
Application Tier

Both the JVM memory and CPU on the application server are utilized efficiently. Tuning of application properties as specified above, gave optimal performance.

Database Servers

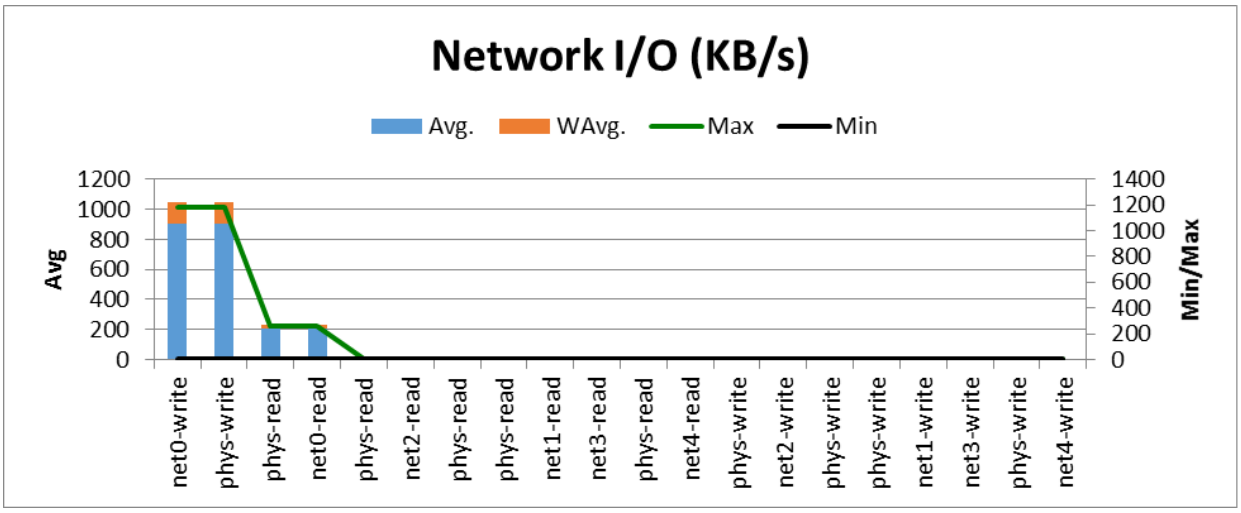
DB Server CPU Utilization - Average – 26.55 %, Max – 40.4%

Below is the graph that shows CPU utilization of a DB server. 26.55 % average utilization has been recorded and went up to 40.4% of maximum utilization.



DB Server Network Utilization

Below is the graph that shows network utilization of a DB server. It can be observed that a series of network reads and writes during the application transaction processing. The average reads are below 201.7 kb/s and the average writes are below 902.1 kb/s. This indicates the network I/O is decent.



Appendix

Test Data Set

Cycle Test Data with a number of activities available for processing is given below:

Transaction Name	Count
PolicyProofOfDeath	20254
BillingStart	22309
Billing	22322
DeathClaimPayout	11300
CoverageCalculation	24391
Commission	20256
DeathBenefitPayout	17540
Issue	42564
DeathNotify	20254
Submit	20255
DeathNotification	11300
AutoTransfer	20255
ADDClaim	11009
DeathClaim	11300
COIPayment	44619
Disbursement	52944
InitialPremium	20255
Premium	44620
Anniversary	22309
BeneficiaryConfirmationRequest	20254
BillingStop	22309
PolicyCancellationRequest	11009
PolicyDeathNotify	20254

Glossary

Term	Description
Application Mode	Determines whether configuration data is cached. When set to PRODUCTION, configuration data is cached.
Application Resource Cache Timeout	Time to cache translations in minutes before checking data source for updates. For value < 0, never check for updates.
Batch Size	The maximum number of Cycle tasks that will process in the cycle grid
Cycle Group Size	The number of cycle work items to group together and execute on a single thread in a cluster.
Cycle Period	The number of seconds that the cycle agent will wait before checking for additional work.
Grid Task Submission Thread Pool Size	The number of threads dedicated to submit tasks to grid.
jpa.showSql (Default – false)	Shows information in the application's log/console for all SQLs executed using JPA. **Note:** It should be used only in a Non-Production environment.
Default Task Processor Threadpoolsize	This thread pool is for processing of activities, scheduled valuation and scheduled computation on any entity that supports such processing. Also this thread pool executes resumable tasks, which are long running tasks in the grid that maintain intermediate state and report progress.
Update stats run	Specifies whether statistics on AsCycle should be updated during policy level of cycle processing. **Values:** Yes, No (default). Example: updatestats.run=Yes
Update stats degree	It specifies the degree of parallelism applicable for Oracle. It is applicable when **updatestats.run=Yes** is set and the value should be set to an integer greater than or equal to 1. If it is not set, then the table default value specified by the DEGREE clause in the CREATE TABLE or ALTER TABLE statement for AsCycle is used. Example: updatestats.degree=32
MEMORY_MAX_TARGET	Specifies the maximum value to which a DBA can set the MEMORY_TARGET initialization parameter.
MEMORY_TARGET	Specifies the Oracle system-wide usable memory. The database tunes memory to the MEMORY_TARGET value, reducing or enlarging the SGA and PGA as needed.

	Note: Settings of MEMORY_MAX_TARGET and MEMORY_TARGET affect each other. In a text-based initialization parameter file, if MEMORY_MAX_TARGET is omitted and a value for MEMORY_TARGET is included, then the database automatically sets MEMORY_MAX_TARGET to the value of MEMORY_TARGET. If a line for MEMORY_TARGET is omitted and a value for MEMORY_MAX_TARGET is included, the MEMORY_TARGET parameter defaults to zero. After startup, the MEMORY_TARGET can be changed dynamically to a nonzero value, provided that it does not exceed the value of MEMORY_MAX_TARGET.
System session-cached-cursors	The session_cached_cursors parameter is used to reduce the amount of parsing with SQL statements that use host variables. The session_cached_cursors parameter has a default value of 50, and increasing the value of session_cached_cursors requires a larger shared_pool_size to cache the cursors.
COMMIT_LOGGING	COMMIT_LOGGING is an advanced parameter used to control how redo is batched by Log Writer. If COMMIT_LOGGING is altered after setting COMMIT_WAIT to FORCE_WAIT, then the FORCE_WAIT option is no longer valid.
COMMIT_WAIT	COMMIT_WAIT is an advanced parameter used to control when the redo for a commit is flushed to the redo logs. If the parameter is set to FORCE_WAIT, the default behavior (immediate flushing of the redo log buffer with wait) is used. If this is a system setting, the session level and transaction level (COMMIT_WRITE) options will be ignored. If this is a session level setting, the transaction level options will be ignored. If COMMIT_WAIT is altered after it has been set, then the FORCE_WAIT option is no longer valid.
CURSOR_SHARING	Determines what kind of SQL statements can share the same cursors. **Values**: » FORCE: Forces the statements that may differ in some literals, but are otherwise, identical to share a cursor unless the literals affect the meaning of the statement. » SIMILAR: Causes statements that may differ in some literals, but are otherwise identical to share a cursor unless the literals affect either the meaning of the statement or the degree to which the plan is optimized. » EXACT: Only allows statements with identical text to share the same cursor.
Number of processes	Specifies the maximum number of operating system user processes that can simultaneously connect to Oracle. Its value should allow for all background processes such as locks, job queue processes, and parallel execution processes. The default values of the SESSIONS and TRANSACTIONS parameters are derived from this parameter. Therefore, if you change the value of PROCESSES, you should evaluate whether to adjust the values of those derived parameters.
Open cursors	Specifies the maximum number of open cursors (handles to private SQL areas) a session can have at once. You can use this parameter to prevent a session from opening an excessive number of cursors.

	It is important to set the value of OPEN_CURSORS high enough to prevent the application from running out of open cursors. The number will vary from one application to another. Assuming that a session does not open the number of cursors specified by OPEN_CURSORS, there is no added overhead to setting this value higher than actually needed.