

Best Practices Guide for Infrastructure Tuning Oracle® Analytics Server (OAS)

*An Oracle® White Paper
May 2022*

Disclaimer:

- All tuning information stated in this guide is for orientation only. Tuning is an iterative process; every modification has to be tested and its impact should be monitored and analyzed.
- Before implementing any of the tuning settings, it is recommended to carry out end-to-end performance testing that will also include obtaining baseline performance data for the default configurations, making incremental changes to the tuning settings and then collecting performance data. Otherwise it may worsen the system performance.

TABLE OF CONTENTS

INTRODUCTION	3
1.0 PERFORMANCE OVERVIEW	3
1.1 INTRODUCTION TO ORACLE® ANALYTICS SERVER PERFORMANCE	3
1.2 PERFORMANCE TERMINOLOGY.....	3
1.3 UNDERSTANDING KEY PERFORMANCE DRIVERS.....	4
2.0 TOP TUNING RECOMMENDATIONS FOR OAS	5
2.1 TUNE OPERATING SYSTEMS PARAMETERS	6
2.1.1 <i>Linux Tuning Parameters</i>	6
2.1.2 <i>Windows Server 2016 / 2019 Tuning Parameters</i>	9
2.2 TUNE ORACLE® WEBLOGIC SERVER (WLS) PARAMETERS	10
2.3 TUNE 64BIT JAVA VIRTUAL MACHINES (JVM)	16
2.3.1 <i>Tuning 64bit Oracle® JVM</i>	16
2.4 TUNE HTTP SERVER PARAMETERS	18
2.4.1 <i>Oracle® HTTP Server (OHS) Tuning</i>	18
2.5 TUNE HTTP SERVER COMPRESSION / CACHING.....	19
2.5.1 <i>Oracle® HTTP Server (OHS)</i>	20
2.7 TUNE DATABASE PARAMETERS.....	22
2.7.1 <i>Web Catalog Objects in Database</i>	22
3.0 PERFORMANCE MONITORING AND TESTING OAS	23
3.1 BUILT-IN BI METRICS FOR PERFORMANCE MONITORING	23
3.2 DIAGNOSING AND RESOLVING ISSUES IN ORACLE ANALYTICS SERVER	26
3.3 PERFORMANCE TESTING ORACLE ANALYTICS SERVER.....	26
3.4 HOW TO ANALYZE HOST SERVER METRICS	27
4.0 TUNING OAS COMPONENTS	30
4.1 ORACLE® BI PRESENTATION SERVICES COMPONENT.....	31
4.2 ORACLE® BI SERVER COMPONENT	32
4.3 ORACLE® JAVAHOST COMPONENT.....	37
4.4 ORACLE® BI SCHEDULERS COMPONENT	39

INTRODUCTION

This document is written for people who monitor performance and tune the components in an Oracle® Analytics environment. It is assumed that readers know server administration, Oracle® Fusion Middleware (FMW), hardware performance tuning fundamentals, web servers, java application servers and database.

1.0 PERFORMANCE OVERVIEW

This chapter discusses performance and tuning concepts for Oracle® Analytics Server. This chapter contains the following sections:

- 1.1 Introduction to Oracle® Analytics Server System Performance
- 1.2 Performance Terminology
- 1.3 Understanding Key Performance Drivers

1.1 Introduction to Oracle® Analytics Server Performance

To maximize Oracle® Analytics Server performance, you need to monitor, analyze, and tune all the Fusion Middleware Oracle® Analytics components. This guide describes the tools that you can use to monitor performance and the techniques for optimizing the performance of Oracle® Analytics Server components.

1.2 Performance Terminology

This guide uses the following performance terminologies:

- **Scalability**
 - System's ability to perform within specification under increasing user load, data load and hardware expansion.
- **Latency**
 - Time between the issuing of a request and the time when the work actually begins on the request.
- **Think time**
 - The time a real user pauses to think between actions.
- **Resource utilization**
 - A consumption metric, for example, the percent of CPU usage.
- **Response time**
 - A time metric, for example round-trip time it takes the server to deliver a Web page.

- **Throughput**

- A rate metric (requests per unit of time), for example, requests per second, bits per second. For example, if an application can handle 20 customer requests simultaneously and each request takes one second to process, this site has a potential throughput of 20 requests per second.

1.3 Understanding Key Performance Drivers

To optimize your deployment, you must understand the elements that influence performance and scalability. A factor that dictates performance is called a key performance driver. Knowing how the drivers behave in combination further enhances your ability to deploy Oracle® Analytics Server optimally, based on the unique requirements of each deployment.

- **Hardware Capacity** - Factors such as quantity of servers, quantity and speed of processors, available RAM, network speed etc.
- **Technical Platforms Tuning** - Fine tuning other third-party software required for installing and running Oracle® Analytics Server product; for example: relational databases, Java application servers, Web servers, Server / Client Operating System and browsers.
- **Business Application Design** - Application design is an important factor in OAS system performance i.e. structure, size, and use of product / custom features in designing reports and dashboards etc.
- **Business process usage** - Activities carried out by users in the normal flow of your business cycle.
 - Business process usage has three components:
 - User activity - Activities available to users for reporting and analysis.
 - Rate of user activity - A number of transactions executed by one user per one hour.
 - User concurrency - Number of users for each activity being carried out simultaneously.

2.0 TOP TUNING RECOMMENDATIONS FOR OAS

Performance tuning Oracle® Analytics Server is a complex iterative process, care needs to be taken to have appropriate backups, proceed incrementally and thoroughly test with each incremental change.

To get you started, we have created a list of recommendations to help you optimize your Oracle® Analytics Server performance.

This chapter includes the following sections that provide a quick start for performance tuning Oracle® Analytics Server infrastructure.

- 2.1 **Tune Operating Systems parameters.**
- 2.2 **Tune Oracle® WebLogic Server (WLS) parameters.**
- 2.3 **Tune 64bit Java Virtual Machines (JVM).**
- 2.4 **Tune HTTP Server parameters.**
- 2.5 **Tune HTTP Server Compression / Caching.**
- 2.6 **Tune Web Browser Settings.**
- 2.7 **Tune Database Parameters.**

Note: While the list in each of the above stated section is a useful tool in starting your performance tuning, it is not meant to be comprehensive list of areas to tune. You must monitor and track specific performance issues within your Oracle® Analytics Server dashboards reports design to understand where tuning can improve performance.

2.1 Tune Operating Systems parameters

Proper OS tuning might improve system performance by preventing the occurrence of error conditions. Operating system error conditions always degrade performance. The following sections describe issues related to operating system performance:

2.1.1 Linux Tuning Parameters

This topic describes how to tune the Linux operating system to optimize the performance of your Oracle® Analytics Server.

Linux Parameters	Default Value	Suggested Value
tcp_fin_timeout By reducing the value of this entry, TCP/IP can release closed connections faster, providing more resources for new connections. Consult your Linux documentation for more information on how to permanently change the value for this parameter.	60	30

Increase Kernel Entropy

On some Linux servers, WebLogic Server Admin / bi_server1 processes startup takes several minutes and hangs for a while. This is normally caused by insufficient low entropy on the Linux server.

a. Check the available entropy:

```
cat /proc/sys/kernel/random/entropy_avail
```

Note: Anything below 500 is at risk of running out of entropy. You can run rngd which replenishes random bytes to /dev/random using /dev/urandom as the source. Start the rngd (as root) and this will ensure your system does not run out of entropy.

```
rngd -r /dev/urandom -o /dev/random -b
```

Ensure that you have installed the rng-tool package on the Linux server:

```
yum -y install rng-tools
```

b. Increase the kernel entropy by adding the following rngd daemon:

1. Navigate to `/etc/init.d`
2. Run `vi /etc/sysconfig/rngd`
3. Make the following changes:

```
EXTRAOPTIONS="-r /dev/urandom -o /dev/random -t 1 -W 4096"
```

Increasing the file descriptor limit

a. Checking open files in Linux

Use the Linux List Open Files (lsof) command to verify the number of open files and network file descriptors that a specific process is using.

The syntax of the lsof command is as follows:

```
lsof -p
```

For example, checking open files for nqsserver:

If you run the *lsof* command immediately after starting the OAS on Linux, it shows that 170 file descriptors were allocated by the nqsserver process (e.g. pid 9219) running on the server. This number (170) is far below the default limit of 1024 file descriptors.

For example:

```
$ lsof -p 9219 | wc -l
170
```

b. Increase the open file limit (*nofile*) and the maximum number of process limit (*nproc*) in the Linux configuration file *limits.d* from 1024 to 131972. For example:

```
[root@bi-1]# cd /etc/security/limits.d/
[root@bi-1 limits.d]# vi 99-oracle.conf
add following and save 99-oracle.conf file:
* soft nofile 131072
* hard nofile 131072
* soft nproc 131072
* hard nproc 131072
```

Note: It is recommended to implement above ulimits settings to avoid following potential OAS system issues:

- If file descriptors (open files - *nofile*) is set to low value, the OAS processes (*sawserver*, *nqsserver*) will stop responding and throw Too Many Open Files error message in logs. You can check current open files value using the "*ulimit -n*" command
- If a max user processes (*nproc*) value is low (default is 1024) the exception *java.lang.OutOfMemoryError: unable to create new native thread* will occur for *bi_serverN/JavaHost* processes. Linux has a maximum allowed process per user limit, that you can check current processes value by using the "*ulimit -u*" command.

Optimize TCP Parameter Settings

Default TCP parameters in most Linux distributions are conservative and are tuned to handle 100Mb/s or 1Gb/s port speed, and result in buffer size that are too small for 10Gb networks. Modifying to below values can lead to significant performance gains in a 10Gb networks link:

```
# Maximum receive socket buffer size
sudo sysctl net.core.rmem_max
# sysctl -w net.core.rmem_max=134217728
```

```
# Maximum send socket buffer size (size of BDP)
sudo sysctl net.core.wmem_max
# sysctl -w net.core.wmem_max=134217728
```

```
# Minimum, initial, and max TCP Receive buffer size in Bytes  
sudo sysctl net.ipv4.tcp_rmem  
# sysctl -w net.ipv4.tcp_rmem="4096 87380 134217728"
```

```
# Minimum, initial, and max buffer space allocated  
sudo sysctl net.ipv4.tcp_wmem  
# sysctl -w net.ipv4.tcp_wmem="4096 65536 134217728"
```

Tip: Allowing more buffer space will have more memory allocated to send/receive sorting data. This can help to improve overall throughput and it will not burst data to the network.

Important Note: Consult your Linux documentation for more information on how to persist above modified kernel values after server reboot.

For more information about Linux tuning, you should consult your Linux documentation. Note the above TCP/kernel tunable parameters and their corresponding values are provided as examples and rough guidelines only. You can re-adjust these, and other parameters based on actual system load, usage patterns such as the number of concurrent users and sessions and so on.

2.1.2 Windows Server 2016 / 2019 Tuning Parameters

This topic describes how to tune the Windows Server 2016 / 2019 operating system to optimize the performance of your Oracle® Analytics Server.

Windows Server 2016 / 2019 Server Parameter	Default Value	Suggested Value
<p>Increase the number of MaxUserPort</p> <p>Under heavy loads it may be necessary to adjust the MaxUserPort. This parameter determines the availability of user ports requested by Oracle® Analytics Server. Tip: By default, the start port is 49152, and the default end port is 65536, this means there are 16384 usable dynamic ports.</p> <p>*Use the following “netsh” command to configure start port and the range:</p> <pre>netsh int ipv4 set dynamicport tcp start=1025 num=64508</pre> <p>Important Note: If you are using firewalls to restrict traffic on your internal network you will need to update the configuration of those firewalls to compensate for the new port range.</p> <p>You can view the current dynamic port range on the server by using the following “netsh” command:</p> <pre>netsh int ipv4 show dynamicport tcp</pre>	16384	64508
<p>Reduce the length of the TcpTimeWaitDelay</p> <p>The TcpTimeWaitDelay registry entry controls the length of time that a connection remains in the <i>TIME_WAIT</i> state before the connection is reused. Reducing this value optimizes the number of connections available for Oracle® Analytics Server processes.</p> <ol style="list-style-type: none">1. Start the registry editor and Navigate to the following entry: <i>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters</i>2. Create a new DWORD (32-bit) Value and Name the value “<i>TcpTimeWaitDelay</i>”.3. Modify the new entry by selecting the decimal radial button.4. Enter 30 for the value and Save the changes.	120	30
<p>Enforce TcpTimeWaitDelay</p> <p>To ensure that the TcpTimeWaitDelay is enforced by the operating system, the value of the StrictTimeWaitSeqCheck should be changed. Note: Decreasing the setting TcpTimedWaitDelay to 30 seconds and setting the StrictTimeWaitSeqCheck to 1 may improve your overall TCP performance.</p> <ol style="list-style-type: none">1. Start the registry editor and Navigate to the following entry: <i>HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters</i>2. Create a new DWORD (32-bit) Value and Name the value “<i>StrictTimeWaitSeqCheck</i>”.3. Modify the new entry and Enter 1 for the value.4. Save the changes.		

2.2 Tune Oracle® WebLogic Server (WLS) parameters

This topic describes how to tune the WebLogic Server to optimize the performance of your Oracle® Analytics Server.

2.2.1 Tuning JDBC Data Sources

You can improve Oracle Analytics Server performance and stability by properly configuring the attributes in JDBC data sources in your *bi* domain.

2.2.1.1 Increase the Number of Connection Pool

If JDBC (BI) data sources are running out of connections to the RCU relational database, then you need to set the maximum number of connections to high value for the following Oracle Analytics Server data sources:

Important Note: The reason why the default value maximum capacity size is set to lower value is because it is important to protect database resources. *So make sure you have enough resources on the database server (available memory is a constraint). If you increase initial / maximum capacity to very high value, you may exceed the limit of PROCESSES / OPEN_CURSORS on Oracle® database server.*

Data Source Name	Default Value	Suggested Value
<i>biplatform_datasource</i>	Initial Capacity: 0 Maximum Capacity: 200	<i>Initial Capacity: 0</i> <i>Maximum Capacity: 400</i>
<i>LocalSvcTblDataSource</i>	Initial Capacity: 0 Maximum Capacity: 200	<i>Initial Capacity: 0</i> <i>Maximum Capacity: 300</i>
<i>mds-owsm</i>	Initial Capacity: 0 Maximum Capacity: 15	<i>Initial Capacity: 0</i> <i>Maximum Capacity: 200</i>
<i>opss-audit-DBDS</i>	Initial Capacity: 0 Maximum Capacity: 15	<i>Initial Capacity: 0</i> <i>Maximum Capacity: 200</i>
<i>opss-audit-viewDS</i>	Initial Capacity: 0 Maximum Capacity: 15	<i>Initial Capacity: 0</i> <i>Maximum Capacity: 200</i>
<i>opss-data-source</i>	Initial Capacity: 0 Maximum Capacity: 15	<i>Initial Capacity: 0</i> <i>Maximum Capacity: 200</i>
<i>WLSSchemaDataSource</i>	Initial Capacity: 1 Maximum Capacity: 75	<i>Initial Capacity: 1</i> <i>Maximum Capacity: 250</i>

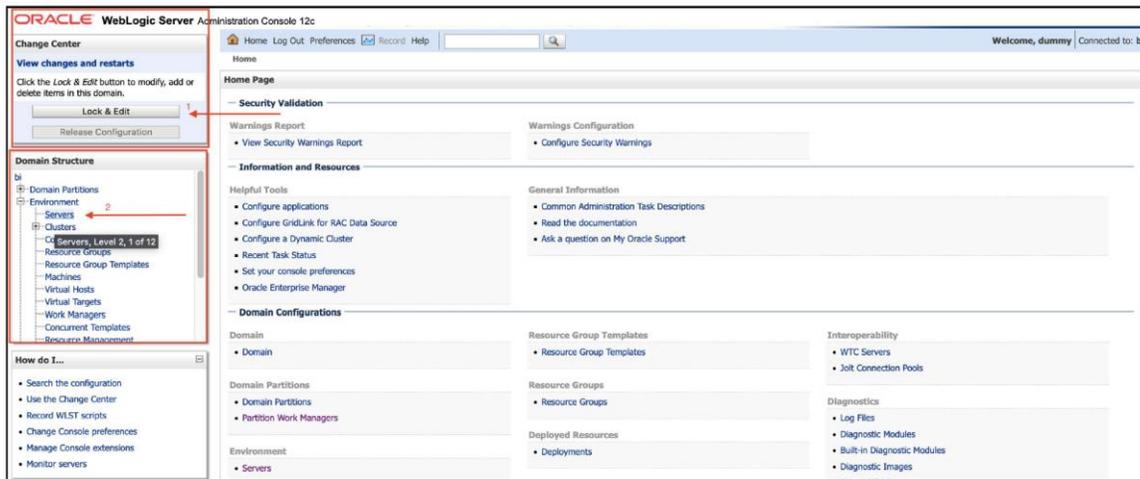
Important Note: *You need to determine the optimal value for the Maximum Capacity as part of your pre-production performance testing. Once optimal values are determined then set the value of Initial Capacity equal to the value for Maximum Capacity to boost performance of JDBC in WebLogic Server web applications.*

2.2.2 Use the `http.keepAliveCache.socketHealthCheckTimeout` system property

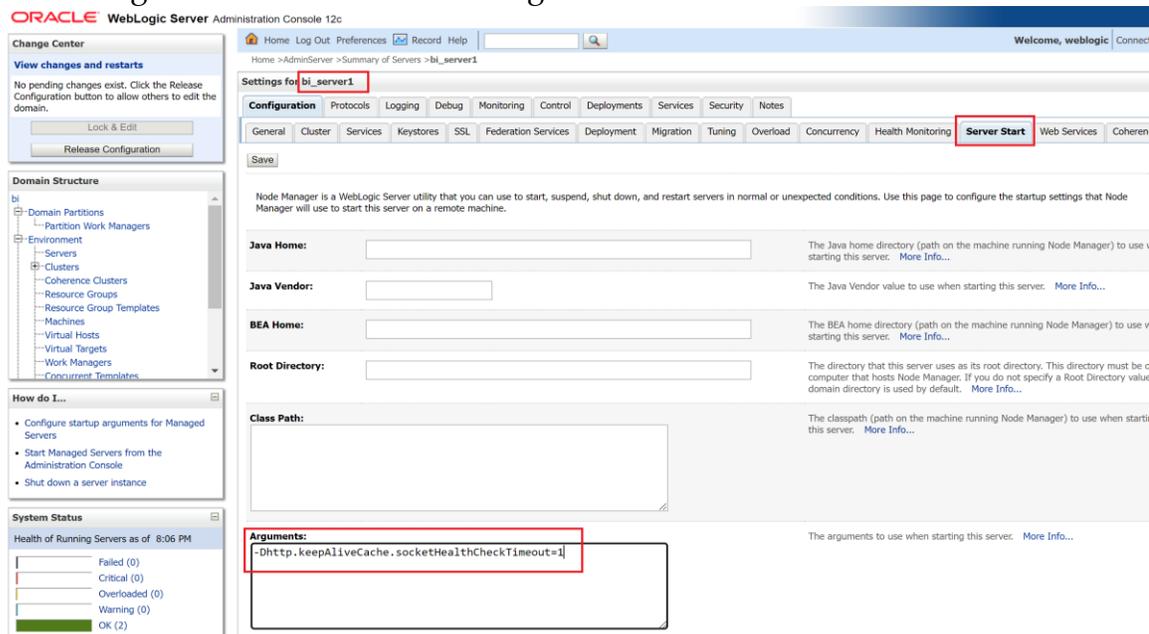
By default, the cache does not check the health condition before returning the cached connection to the client for use. Under some conditions, such as due to an unstable network connection, the system needs to check the connection's health condition before returning it to the client. To enable this behavior (checking the health condition), set `-Dhttp.keepAliveCache.socketHealthCheckTimeout` to 1.

Tip: Add above `-Dhttp.keepAliveCache.socketHealthCheckTimeout` setting, you need to:

1. Access the WebLogic Admin Console. On the top left side of the login page, under the Change Center click the "Lock & Edit" on the left side pane.
2. Next, under the domain structure, click the servers:



3. Under Summary of Servers, click on `bi_server1` and navigate to the "server start" tab and add the flag `-Dhttp.keepAliveCache.socketHealthCheckTimeout=1` in 'Arguments:' and save the changes as shown below



4. Note: If you have clustered setup then repeat the same steps to add the flag "*-Dhttp.keepAliveCache.socketHealthCheckTimeout=1*" for the rest of the bi_serverN nodes e.g. bi_server1, bi_server2, bi_server3...
5. Finally, click the Release Configuration under the Change Center on the top left.
6. Restart the bi_serverN managed server from location i.e. *../bi/bitools/bin/stop.sh -i bi_server1* and *../bi/bitools/bin/start.sh -i bi_server1*
7. After the restart is completed, check if the server start arguments are applied. To verify, check the bi_server1 process and look for parameter > "*-Dhttp.keepAliveCache.socketHealthCheckTimeout=1*".

2.2.3 Stuck thread detection behavior Tuning

WebLogic Server automatically detects when a thread in an execute queue becomes "stuck." Because a stuck thread cannot complete its current work or accept new work, the server logs a message each time it diagnoses a stuck thread.

A thread might get stuck due to various reasons. For example: When large BI report is running and the time it takes to complete is say 800 seconds, then, as the default stuck thread timing is 600 seconds in WebLogic Server, the thread allocated for that query waits for 600 seconds and goes to the stuck state.

Note: the stuck state does not mean that it is stuck forever, and it might get unstuck at a later point in time which can be checked in the BI managed server logs. But if there is no reference like that then it might infer that the thread is stuck forever

Use the following Stuck Threads suggested settings as a starting point to improve stuck thread detection, and then, after careful testing of your long running BI transactions, adjust as needed.

Managed Servers	Default Value	Suggested Value
bi_server1	Stuck Thread Max Time=600 Stuck Thread Timer Interval=60 Max Stuck Thread Time=600	<i>Stuck Thread Max Time=3600</i> <i>Stuck Thread Timer Interval=3600</i> <i>Max Stuck Thread Time=3600</i>
AdminServer	Stuck Thread Max Time=600 Stuck Thread Timer Interval=60 Max Stuck Thread Time=600	<i>Stuck Thread Max Time=3600</i> <i>Stuck Thread Timer Interval=3600</i> <i>Max Stuck Thread Time=3600</i>

Tip: To configure the above stuck thread detection behavior settings, you need to:

- Access the Administration Console for the BI domain.
- Expand the Servers node in the left pane to display the servers configured in your BI domain.
- Click on the BI Managed Server (*bi_serverN*) instance and then select the Configuration -> Tuning tab in the right pane. Modify the attributes i.e. Stuck Thread Max Time and Stuck Thread Timer Interval
- Then click on the Overload tab for each BI Managed Server (*bi_serverN*) and modify the Max Stuck Thread Time attribute.

2.2.4 To improve the performance of WebLogic and LDAP Authentication providers: To optimize the group membership caches for WebLogic and LDAP Authentication providers, set the following attributes (found in the Administration Console on the *LDAP Authentication provider's Configuration* → *Provider Specific and Performance* pages):

- **Group Membership Searching**—Controls whether group searches are limited or unlimited in depth. This option controls how deeply to search into nested groups. For configurations that use only the first level of nested group hierarchy, this option allows improved performance during user searches by limiting the search to the first level of the group.
 - If a limited search is defined, Max Group Membership Search Level must be defined.
 - If an unlimited search is defined, Max Group Membership Search Level is ignored.
- **Max Group Membership Search Level**—Controls the depth of a group membership search if Group Membership Searching is defined. Possible values are:
 - 0 (zero) - Indicates only direct groups will be found. That is, when searching for membership in Group A, only direct members of Group A will be found. If Group B is a member of Group A, the members will not be found by this search.
 - Any positive number—indicates the number of levels to search. For example, if this option is set to 1, a search for membership in Group A will return direct members of Group A. If Group B is a member of Group A, the members of Group B will also be found by this search. However, if

Group C is a member of Group B, the members of Group C will not be found by this search.

Tip: *If you see BI Managed Server JVM segfaults and then it restarts / hangs when a user logs in, check the user group memberships in Active Directory. If the user belongs to groups in a circular reference (group A belongs to group B that belongs to group A), without additional configuration parameters, the JVM may segfaults. To workaround that set in WebLogic Server Admin console the "Max Group Membership Search Level" for the AD provider. For more information, refer to "Improving the Performance of WebLogic and LDAP Authentication Providers" section at https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/secmg/ldap_atn.html#GUID-E50EFFD3-51EB-4D0A-BC1D-6BB08C73D68E*

2.2.5 Tune LibOVD searches

LibOVD is a java library providing virtualization capabilities over LDAP authentication providers in Oracle Fusion Middleware. LibOVD is activated when you set the property *virtualize=true* for the identity store provider in *jps-config.xml*.

Setting the libOVD property attribute parameter *OPTIMIZE_SEARCH=true* will improve the performance of searches as it forces libOVD to search only within the users and groups search bases defined in the authenticator providers. No searches are performed elsewhere.

Tip: *To add libOVD property OPTIMIZE_SEARCH=true via Enterprise Manager refer to doc at Managing Security in <https://docs.oracle.com/en/middleware/bi/analytics-server/security-oas/configure-oracle-analytics-server-use-alternative-authentication-providers.html#GUID-0EDD6411-21BE-4C90-8337-88ADF97FEF11>*

2.2.6 Tune WebLogic Server Node Manager Java Heap Size

In Oracle Analytics Server, WebLogic Server Node Manager is monitoring not only WebLogic Managed Servers but also BI System components. When starting all the BI processes using start.sh, Node Manager may take longer time to start, so it is recommended to increase the java heap size for Node Manager.

File "*commBaseEnv.sh*" containing the Node Manager Java Heap is located at `<ORACLE_HOME>/oracle_common/common/bin`

Update the following *MEM_ARGS* parameter from default of "*-Xms32m -Xmx200m*" to higher values according to available physical memory on the system:

```
...
else
case $JAVA_VENDOR in
Oracle)
if [ "${VM_TYPE}" = "JRockit" ]; then
JAVA_VM=-jrockit
MEM_ARGS="-Xms128m -Xmx256m"
UTILS_MEM_ARGS="-Xms32m -Xmx1024m"
else
JAVA_VM=-server
MEM_ARGS="-Xms32m -Xmx200m"
UTILS_MEM_ARGS="-Xms32m -Xmx1024m"
fi
VERIFY_NONE="-Xverify:none"
...
to
...
else
case $JAVA_VENDOR in
Oracle)
if [ "${VM_TYPE}" = "JRockit" ]; then
JAVA_VM=-jrockit
MEM_ARGS="-Xms128m -Xmx256m"
UTILS_MEM_ARGS="-Xms32m -Xmx1024m"
else
JAVA_VM=-server
MEM_ARGS="-Xms2048m -Xmx4096m"
UTILS_MEM_ARGS="-Xms32m -Xmx1024m"
fi
VERIFY_NONE="-Xverify:none"
...
```

Note: Verify in "*commBaseEnv.sh*" file that for all the *JAVA_VM=-jrockit* is changed from *MEM_ARGS="-Xms32m -Xmx200m"* to ***MEM_ARGS="-Xms2048m -Xmx4096m"***. Save the file and restart the services (using stop.sh & start.sh located at `<ORACLE_HOME>/user_projects/domains/bi/bitools/bin`)

2.3 Tune 64bit Java Virtual Machines (JVM)

Newly deployed web application server instances use default memory heap settings, which are often too small to accommodate Oracle Analytics Server 64-bit requirements. Optimal JVM settings can be determined only by close monitoring of application server performance under peak realistic load.

2.3.1 Tuning 64bit Oracle® JVM

The table below lists the memory settings that applies to OAS managed servers in bi domain i.e. *bi_serverN* and *AdminServer*.

Important Note: Below the suggested value, set minimum and maximum to the same value (*8g in this example – depending on available physical memory on each server, you need to enter the appropriate value for your system*). Making them the same size means the JVM will not spend time trying to work out whether it needs to increase the size of the heap:

Settings	Suggested Value	Java Argument*
Minimum heap (bi_server1)	8g	-Xms8g
Maximum heap (bi_server1)	8g	-Xmx8g
Minimum heap (AdminServer)	1024 MB	-Xms1024m
Maximum heap (AdminServer)	4096 MB	-Xmx4096m

***Warning:** Make sure you have enough physical memory allocated available on the servers and then use the above settings as a starting point, and after load testing, adjust as needed. For a stable even performance over time, you should set the initial heap size (-Xms) to the same value as the maximum heap size (-Xmx).

Tip: How to increase the Oracle JVM heap memory value for OAS managed servers (bi_serverN and AdminServer):

1. bi_serverN heap size:

- Browse to `<Oracle_Home>/user_projects/domains/bi/bin` and backup the file `setStartupEnv.sh`.
- Edit `setStartupEnv.sh` and update to the following lines highlighted in bold inside `"${STARTUP_GROUP}" = "BISUITE-MAN-SVR"` section:

64 bit JVM memory settings

```
SERVER_MEM_ARGS_64="-Xms8g -Xmx8g"  
export SERVER_MEM_ARGS_64  
SERVER_MEM_ARGS_64HotSpot="-Xms8g -Xmx8g"  
export SERVER_MEM_ARGS_64HotSpot  
SERVER_MEM_ARGS_64JRockit="-Xms8g -Xmx8g"  
export SERVER_MEM_ARGS_64JRockit
```

2. AdminServer heap size:

- Browse to <Oracle_Home>/user_projects/domains/bi/bin and backup the file *setStartupEnv.sh* .
- Edit *setStartupEnv.sh* and update to the following lines highlighted in bold inside "*\${STARTUP_GROUP}*" = " *AdminServerStartupGroup*" section:

64 bit JVM memory settings

```
SERVER_MEM_ARGS_64="-Xms1024m -Xmx4096m"  
export SERVER_MEM_ARGS_64  
SERVER_MEM_ARGS_64HotSpot="-Xms1024m -Xmx4096m"  
export SERVER_MEM_ARGS_64HotSpot  
SERVER_MEM_ARGS_64JRockit="-Xms1024m -Xmx4096m"  
export SERVER_MEM_ARGS_64JRockit
```

2.4 Tune HTTP Server parameters

This topic describes how to tune the HTTP server to optimize the performance of your Oracle® Analytics Server.

2.4.1 Oracle® HTTP Server (OHS) Tuning

Oracle® HTTP Server uses directives in *httpd.conf* file. The directives for each Multi-Processing Module (MPM) type are defined in the *httpd.conf* file. The default MPM type is *Worker MPM*.

Parameters	Default Value	Suggested Value
MaxKeepAliveRequests	100	0
KeepAliveTimeout	5	120
KeepAlive	On	On
Timeout	300	6000
# WinNT MPM (this MPM is for Windows Only)		
<i><IfModule mpm_winnt_module></i>		
<i> ThreadsPerChild</i>	= 150	= 8096
<i> ListenBackLog</i>	= Default	= 8096
<i> MaxRequestsPerChild</i>	= 0	= 0
<i></IfModule></i>		
#worker MPM (by default OHS use multithreaded mode in UNIX)		
<i>#worker MPM</i>		
<i><IfModule mpm_worker_module></i>		
<i>StartServers 3</i>		
<i>ServerLimit 200</i>		
<i>MinSpareThreads 450</i>		
<i>MaxSpareThreads 750</i>		
<i>ThreadsPerChild 64</i>		
<i>MaxRequestWorkers 1500</i>		
<i>MaxConnectionsPerChild 0</i>		
<i>Mutex fcntl:\${ORACLE_INSTANCE}/servers/\${COMPONENT_NAME}/logs</i>		
<i></IfModule></i>		
<i># event MPM</i>		
<i><IfModule mpm_event_module></i>		
<i>StartServers 3</i>		
<i>ServerLimit 200</i>		
<i>ThreadLimit 250</i>		
<i>MinSpareThreads 450</i>		
<i>MaxSpareThreads 750</i>		
<i>ThreadsPerChild 64</i>		
<i>MaxRequestsPerChild 0</i>		
<i>MaxRequestWorkers 1500</i>		
<i>MaxConnectionsPerChild 0</i>		
<i>AsyncRequestWorkerFactor 2</i>		
<i>Mutex fcntl:\${ORACLE_INSTANCE}/servers/\${COMPONENT_NAME}/logs</i>		
<i></IfModule></i>		

2.5 Tune HTTP Server Compression / Caching

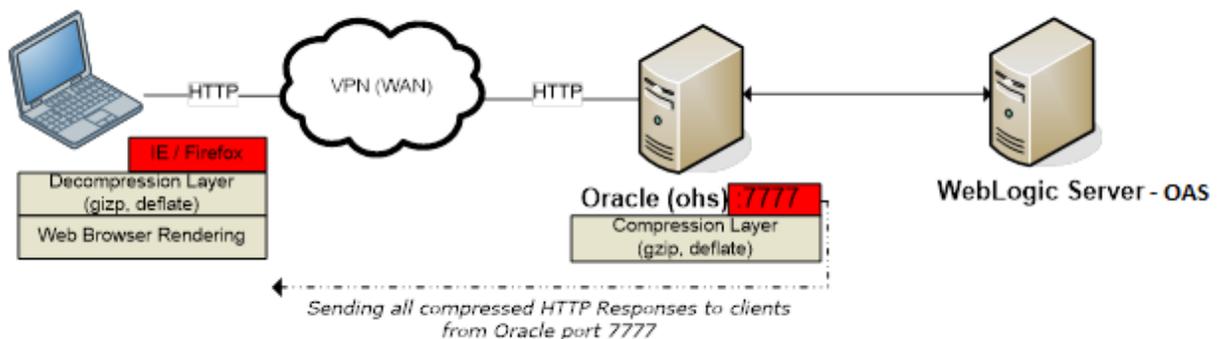
Why use Web Server Compression / Caching for Oracle® Analytics Server?

1. Bandwidth Savings: Enabling HTTP compression can have a dramatic improvement on the latency of responses. Compressing static files and dynamic application responses will significantly reduce remote (high latency) user response time.
2. Improves request/response latency: Caching makes it possible to suppress the payload of the HTTP reply using the 304-status code. *Minimizing round trips over the Web to revalidate cached items can make a huge difference in browser page load times.*

2.6.0.1 Web Server Compression Flow

To better understand compression flow, the illustration below depicts the flow and where the compression and decompression occurs on Oracle® HTTP Server (OHS).

Compression enabled on Oracle® HTTP server (OHS) level:



2.5.1 Oracle® HTTP Server (OHS)

This topic describes how to enable caching / compression in Oracle® HTTP Server of your Oracle® Analytics Server. **Important Note:** High load of HTTP replies with 304 status code causes the Oracle Analytics Server UI to work slowly in browser over high latency networks. To resolve this issue, it is highly recommended to implement HTTP caching and compression which will minimize the round trips over the Web to revalidate cached items, resulting in a huge difference in browser page load times.

a. How to Enable Compression and Caching:

1. To implement HTTP compression / caching, install and configure Oracle HTTP Server (OHS) for the Oracle Analytics Server (refer to "Enterprise Deployment Guide for Oracle Analytics Server" document for details: <https://docs.oracle.com/en/middleware/bi/analytics-server/enterprise-deploy-oas/enterprise-deployment-overview.html>).

2. On the OHS machine, open the file HTTP Server configuration file (*httpd.conf*) for editing. This file is in the OHS installation directory. For example: *./user_projects/domains/base_domain/config/fmwconfig/components/OHS/instances/ohs1*

3. In *httpd.conf* file, verify that the following directives are included and not commented out:

```
LoadModule expires_module "${PRODUCT_HOME}/modules/mod_expires.so"
LoadModule deflate_module "${PRODUCT_HOME}/modules/mod_deflate.so"
```

4. Add the following lines in *httpd.conf* file below the directive *LoadModule* section to compression / caching and restart the OHS:

```
#HTTP Compression
<IfModule mod_deflate.c>
    SetOutputFilter DEFLATE
</IfModule>

#HTTP Caching of static files
ExpiresActive On
<IfModule mod_expires.c>
    ExpiresByType image/gif "access plus 6 months"
    ExpiresByType image/jpeg "access plus 6 months"
    ExpiresByType application/x-javascript "access plus 6 months"
    ExpiresByType text/css "access plus 6 months"
    ExpiresByType text/javascript "access plus 6 months"
    ExpiresByType image/png "access plus 6 months"
</IfModule>

#This stops the HTTP 304 replies in browser
<IfModule mod_headers.c>
    <FilesMatch "\.(gif|jpeg|png|x-javascript|javascript|css|swf)$">
        Header set Cache-Control "max-age=7889231"
    </FilesMatch>
</IfModule>
```

b. Performance Gain Test (Oracle® HTTP Server)

The test with/without HTTP compression enabled was conducted to measure the transactions response time / throughput for Oracle® Analytics Server.

The table below summarizes the performance improvement for OAS transactions over a 1 Mbps bandwidth link for one remote user.

Transaction Name	Average time (in sec)	Average time (in sec)	% Improvement
	Before (Non-Compressed)	After (Compressed)	
Total transactions response time: <i>OAS Homepage > Display report (with 1000 rows) > Click & Navigate to next 1000 rows page up to 5 times.</i>	300 seconds	20 seconds	93%
Total Throughput (bytes) <i>(Total generated network traffic for one user)</i>	18,534,557	1,123,646	94%
Average Network Delay (ms)	350	355	-
Packet Loss %	0	0	-
HTTP Response Codes Count	HTTP/200: 483	HTTP/200: 483	-

2.7 Tune Database Parameters

Tuning and indexing underlying databases: For Oracle BI Server database queries to return quickly, the underlying databases must be configured, tuned, and indexed correctly. Note that different database products have different tuning considerations.

Tip: If there are queries that return slowly from the underlying databases, then you can capture the SQL statements for the queries in the query log and provide them to the database administrator (DBA) for analysis. See "Managing the Query Log" in the *Administering Oracle Analytics Server* guide (<https://docs.oracle.com/en/middleware/bi/analytics-server/administer-oas/manage-query-log.html>) for more information about configuring query logging on the system.

2.7.1 Web Catalog Objects in Database

Catalog objects are now stored in database tables rather than the file system. In Oracle Analytics Server, catalog objects are stored in database tables that are created by the Repository Creation Utility (RCU).

To ensure consistent optimal performance of catalog objects metadata calls following database deployment is recommended where RCU schemas are hosted:

- Network latency between OAS application servers and RCU schemas hosted Database server should be less than < 1ms.
- Database server where RCU schemas are hosted should be well fine-tuned as per the Database Performance Tuning Guide.

3.0 PERFORMANCE MONITORING AND TESTING OAS

This topic outlines the processes that have to be monitored for the Oracle® Analytics Server using built-in BI metrics and default Operating System performance utilities.

3.1 Built-in BI Metrics for Performance Monitoring

You can use the following built-in BI metrics to monitor System Components (OAS processes) and WebLogic Server processes.

Tips:

- Use ODBC/JDBC Procedures to Obtain BI Server Diagnostics, for more details refer to <https://docs.oracle.com/en/middleware/bi/analytics-server/administer-oas/use-odbc-jdbc-procedures-obtain-bi-server-diagnostics.html>
- Capture Metrics Using the Dynamic Monitoring Service, for more details refer to <https://docs.oracle.com/en/middleware/bi/analytics-server/administer-oas/capture-metrics-using-dynamic-monitor-service.html#GUID-989AA8FA-5129-42AB-90D9-072951800DBB>
- Use Oracle Enterprise Manager (EM) URL to monitor end to end OAS real time performance: <http://<server>:9500/em> . For more details refer to <https://docs.oracle.com/en/middleware/bi/analytics-server/administer-oas/manage-system-components-using-fusion-middleware-control.html#GUID-9D28BB9F-E00E-4AB0-A7E5-BB17869F9CD0>
- In Oracle Analytics Server, the *perfmon* URL is still valid to use i.e. <http://<server:port>/analytics/saw.dll?Perfmon>
- Oracle Analytics Server from version 6.4 onwards includes performance tools for Data Visualization, for more details refer to [Developer Reference](#)

3.1.1 How to Analyze Oracle Analytics Server Metrics

If you encounter a problem, such as Presentation Services and BI Server components running slowly or hanging, you can view more detailed performance information, including performance metrics for a particular target to further diagnose the problem.

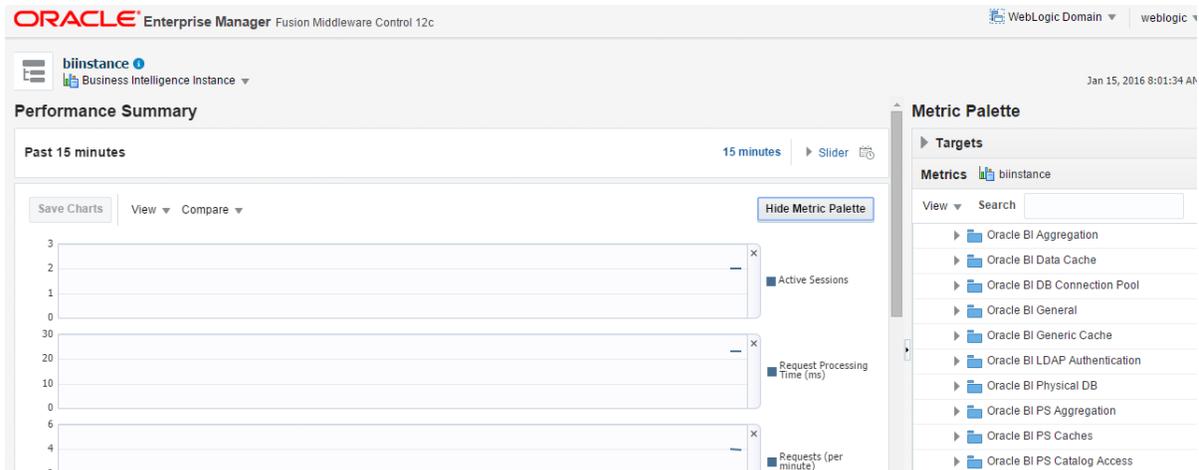
Tip: To view the performance of an Oracle Presentation Services and BI Server:

1. From the navigation pane, expand the farm, then **Business Intelligence**, and then the **biinstance**.
The Business Intelligence instance home page is displayed.
2. From the Business Intelligence instance menu, choose **Monitoring > Performance Summary**:

The Performance Summary page is displayed. It shows performance metrics, as well as information about Active Sessions, Current Sessions, Total sessions and Queries/sec for OBIPS and Oracle BI Server.

3. To see additional metrics, click *Show Metric Palette* and expand the metric categories.

The following figure shows the Performance Summary page with the Metric Palette displayed:



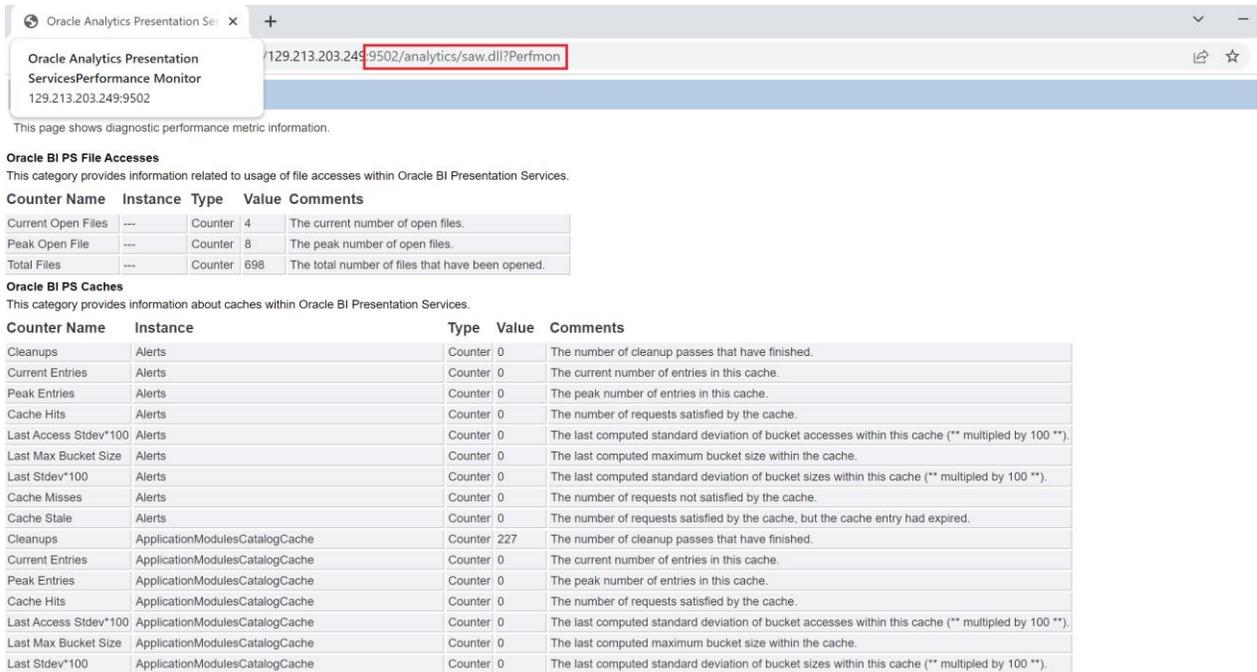
1. Select a metric to add it to the Performance Summary.
2. To overlay another target, click *Overlay > Another BI Instance...*, and select the target. The target is added to the charts, so that you can view the performance of more than one target at a time, comparing their performance.
3. To customize the time frame shown by the charts, you can:
 - Click the **Slider** to display a slider tool that lets you define the duration in the charts. For example, to show the past 10 minutes, instead of the past 15 minutes, slide the left slider control to the right until it displays the last 10 minutes.
 - Select the calendar and clock icon. Then, enter the **Start Time** and **End Time**.

Tip: You can also view the performance of components, such as Oracle HTTP Server or Oracle WebLogic Server. Navigate to the component and select **Monitoring**, then **Performance Summary** from the dynamic target menu.

3.1.2 Viewing Oracle Presentation Services perfmon page

In your web browser, type in *http://<server:port>/analytics/saw.dll?Perfmon*

Note: You need to login with BI Administrator role.



The screenshot shows a web browser window with the URL `http://129.213.203.249:9502/analytics/saw.dll?Perfmon`. The page title is "Oracle Analytics Presentation ServicesPerformance Monitor". Below the title, there is a section for "Oracle BI PS File Accesses" and a table with columns: Counter Name, Instance, Type, Value, and Comments. The table lists metrics like Current Open Files, Peak Open File, and Total Files. Below that is a section for "Oracle BI PS Caches" and another table with columns: Counter Name, Instance, Type, Value, and Comments. This table lists various cache-related metrics such as Cleanups, Current Entries, Peak Entries, Cache Hits, Last Access Stdev*100, Last Max Bucket Size, Last Stdev*100, Cache Misses, and Cache Stale.

Counter Name	Instance	Type	Value	Comments
Current Open Files	---	Counter	4	The current number of open files.
Peak Open File	---	Counter	8	The peak number of open files.
Total Files	---	Counter	698	The total number of files that have been opened.

Counter Name	Instance	Type	Value	Comments
Cleanups	Alerts	Counter	0	The number of cleanup passes that have finished.
Current Entries	Alerts	Counter	0	The current number of entries in this cache.
Peak Entries	Alerts	Counter	0	The peak number of entries in this cache.
Cache Hits	Alerts	Counter	0	The number of requests satisfied by the cache.
Last Access Stdev*100	Alerts	Counter	0	The last computed standard deviation of bucket accesses within this cache (** multiplied by 100 **).
Last Max Bucket Size	Alerts	Counter	0	The last computed maximum bucket size within the cache.
Last Stdev*100	Alerts	Counter	0	The last computed standard deviation of bucket sizes within this cache (** multiplied by 100 **).
Cache Misses	Alerts	Counter	0	The number of requests not satisfied by the cache.
Cache Stale	Alerts	Counter	0	The number of requests satisfied by the cache, but the cache entry had expired.
Cleanups	ApplicationModulesCatalogCache	Counter	227	The number of cleanup passes that have finished.
Current Entries	ApplicationModulesCatalogCache	Counter	0	The current number of entries in this cache.
Peak Entries	ApplicationModulesCatalogCache	Counter	0	The peak number of entries in this cache.
Cache Hits	ApplicationModulesCatalogCache	Counter	0	The number of requests satisfied by the cache.
Last Access Stdev*100	ApplicationModulesCatalogCache	Counter	0	The last computed standard deviation of bucket accesses within this cache (** multiplied by 100 **).
Last Max Bucket Size	ApplicationModulesCatalogCache	Counter	0	The last computed maximum bucket size within the cache.
Last Stdev*100	ApplicationModulesCatalogCache	Counter	0	The last computed standard deviation of bucket sizes within this cache (** multiplied by 100 **).

3.1.3 Using Usage Tracking Statistics

The Oracle BI Server supports the accumulation of usage tracking statistics that can be used in a variety of ways such as database optimization, aggregation strategies, or billing users or departments based on the resources that they consume.

The BI Server tracks usage at the detailed query level. It is recommended to use production usage tracking data to design the workload for your performance load testing. See "[Track Usage](#)" in *Oracle® Administering Oracle Analytics Server guide*.

3.2 Diagnosing and Resolving Issues in Oracle Analytics Server

The usual indication that you should troubleshoot Oracle BI will be sluggish performance of BI component. Examples of BI components are charts, tables, dashboards, and queries. Many configuration performance issues can be detected from the following logs:

- **Presentation Services Log** (*sawlog<n>.log*).
- **BI Server Log** (*obis<n>_query.log*, *obis1-diagnostic<n>.log*).
- **Scheduler Log** (*nqscheduler.log*).
- **JavaHost Log** (*jh.log*).
- **Cluster Controller Log** (*nqcluster.log*).
- **WLS Managed Servers Log** (*AdminServer-diagnostic.log*, *AdminServer.log*, *bi_server1.log*, *bi_server1-diagnostic.log*).

Important Note: For additional information about Diagnosing and Resolving Issues in Oracle Analytics Server, see “[Diagnose and Resolve Issues](#)” in Oracle® Administering Oracle Analytics Server guide.

3.3 Performance Testing Oracle Analytics Server

Oracle Support Document 2518309.1 (Load Testing Oracle Analytics Cloud (OAC) Using Oracle Load Testing (OLT) 13.x) can be found at: <https://support.oracle.com/epmos/faces/DocumentDisplay?id=2518309.1> and is intended as a guide for creating load/performance test scripts against Oracle Analytics Server using Oracle Load Testing 13.x.

This guide will assist during the script creation process and enable the OAS tester to create scripts faster and more reliably. It assumes that the person using this document has experience working with Oracle Application Testing Suite.

Note: This support document (*Doc ID 2518309.1*) does not necessarily cover all OAS transactions that have to be tested, nor does it guarantee that the parameters mentioned in this document will perfectly match your particular environment.

3.4 How to Analyze Host Server Metrics

Administrators will find it useful to study these suggestions to undertake when a metric value exceeds threshold. The commands provided are for the Linux and Windows operating system.

When logical free memory/swap activity or paging activity is beyond threshold i.e. the combined value of Pages (Paged-in and Pages Paged-out) should be <=1000:

This usually happens when memory is not sufficient to handle demands from all the running processes.

- Linux: Check “*cat /proc/meminfo*” and confirm the total RAM expected.
Windows: Open the Task Manager, click the Performance tab and check the Physical Memory section.
- Check whether there are unallocated huge pages. If there are and the WebLogic Server / Oracle BI instances (OBIPS, JH, OBIS) are not expected to use them, reduce the huge page pool size.
- Linux: Run *top* and sort by resident memory (type OQ). Look for processes using the most resident memory and investigate those processes.
Windows: Open the Task Manager, click the Processes tab and click the Mem Usage column to sort the processes by memory usage

When Network Interface Error Rates Is Beyond Threshold

The normal cause for this is a misconfiguration between the host and the network switch. A bad network card or cabling also can cause this error. You can run */sbin/ifconfig* to identify which interface is having packet errors. Contact network administrator to ensure the host and the switch are using same data rate and duplex mode.

Otherwise, check whether cabling or the network card is faulty and replace as appropriate.

When Packet Loss Rate Is Beyond Threshold

The normal cause of this error is network saturation of bad network hardware.

- Run *lsof -Pni | grep ESTAM* to determine which network paths are generating the problem. Then run *mtr <target host>* or *ping <target host>* and look for packet lost on that segment.

```
20 packets transmitted, 20 received, 0% packet loss, time 18997ms
```

rtt min/avg/max/mdev = 0.168/0.177/0.200/0.010 ms

The packet loss should be 0% and rtt should be less than .5 ms.

- Ask the network monitoring staff to look for saturation or network packet loss from their side.

When Network Utilization Is Beyond Threshold i.e. All Network Interfaces Combined Utilization > 95%

The normal cause is very heavy application load.

- Run top or lsof to determine which processes are moving a lot of data.
- Use tcpdump to sample the network for usage patterns.
- Use atop, iftop, ntop or pkstat to see which processes are moving data.

When CPU Usage or Run Queue Length Is Beyond Threshold i.e. Run Queue (5 min average) > 4 (The run queue is normalized by the number of CPU cores):

The normal cause is runaway demand, a poorly performing application, or poor capacity planning.

- Linux: Run top to identify which application/process is using time.
Windows: Open the Task Manager, click the Processes tab and click the CPU column to sort the processes based on CPU usage.
- If top processes are WebLogic Server JVM processes, conduct a basic WebLogic Server health check. That is, review logs to see if there are configuration errors causing excessive exceptions, and review metrics to see if the load has increased.
- If top processes are Oracle BI processes, use Oracle Enterprise Manager Control to look for BI components level statistics.

When System CPU Usage Is Beyond Threshold i.e. CPU Utilization > 95%

- High system CPU use could be due to kernel processes looking for pages to swap out during a memory shortage.
- High system CPU use is also frequently related to various device failures. Run `dmesg | less` and look for repeated messages about errors on some particular device, and also have hardware support personnel check the hardware console to see if there are any errors reported.

When Filesystem Usage Is Beyond Threshold i.e. Filesystem Space Available <5%

The normal cause is an application that is logging excessively or leaving behind temporary files.

- Run `lsdf -d 1-99999 | grep REG | sort -nrk 7 | less` to see currently open files sorted by size from largest to smallest. Investigate the large files.
- Run `du -k /mount_point_running_out_of_space > /tmp/sizes` to get space used for directories under the mount point. This may take a long time. While it is running, run `sort -nr /tmp/sizes` and find the directories using most space and investigate those first.

When Total Processes Is Beyond Threshold > 25000

The normal cause is runaway code or a stuck NFS filesystem.

- Linux: Run `ps aux`. If many processes are in status D, run `df` to check for stuck mounts.
Windows: Run Task Manager, click the Processes tab, and check the list of running processes. If there are hundreds or thousands of processes of a particular program, determine why.
- Run `ps o pid,nlwp,cmd | sort -nrk 2 | head` to look for processes with many threads.

When Disk Device Busy Is Beyond Threshold > 95%

- Check for disk drive failure.

Linux: As root, check `/var/log/messages*` and `/var/log/mcelog` to see if there are any error messages indicating disk failure. For a RAID array, the disk controller needs to be checked. The commands will be specific to the controller manufacturer.

Windows: Run `perfmon` and look at the Alert logs. Run `chkdsk` to check for disk failure.

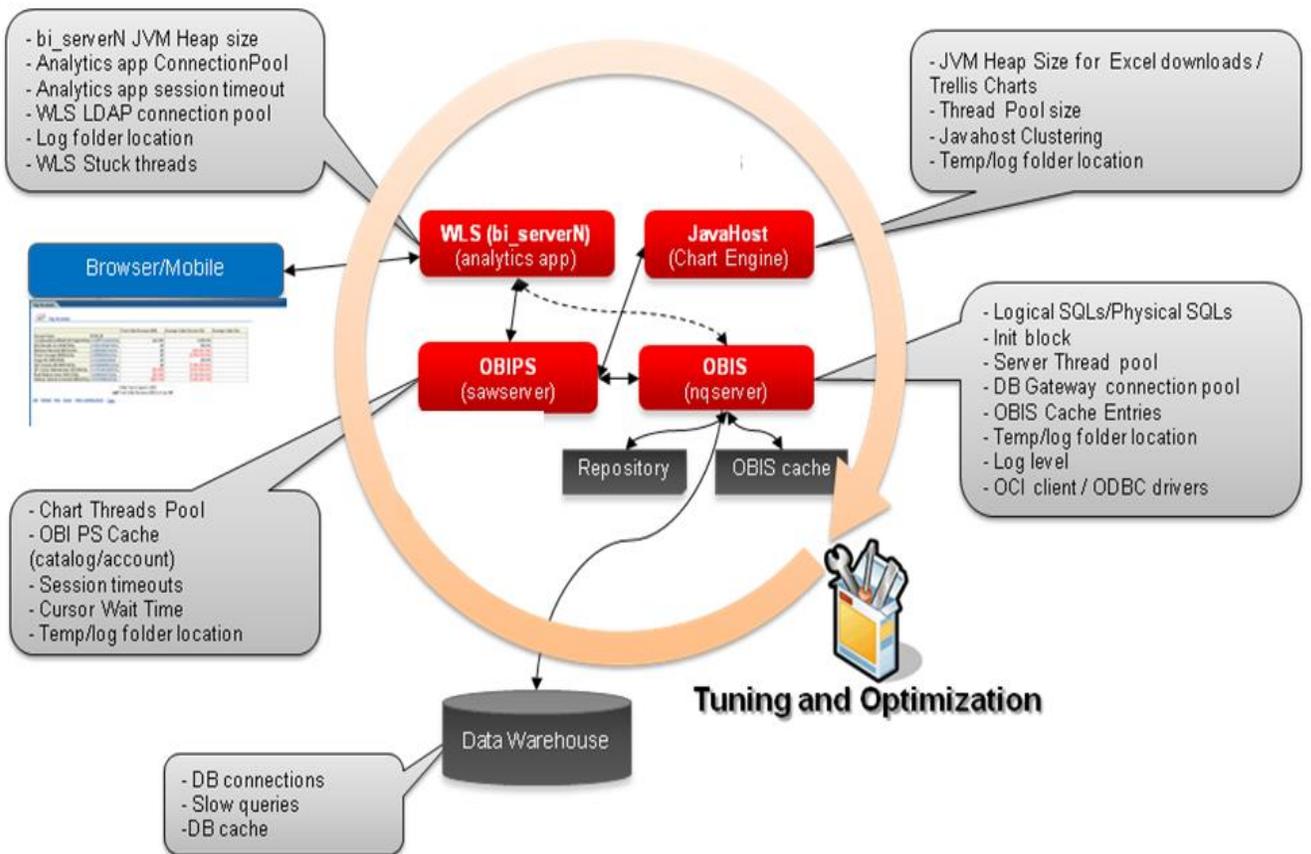
- Look for processes that are using the disk. From a shell window, execute `ps aux | grep 'D.'` several consecutive times to look for processes with "stat" D.

4.0 TUNING OAS COMPONENTS

This chapter provides a quick start guide for tuning main Oracle® Analytics Server system components (i.e. BI Presentation Services, JavaHost, BI Server).

4.0.1 OAS Performance Tuning Map

Below OAS performance tuning map can help you to navigate the main OAS performance components that need to be tuned and help you to resolve your BI performance issues.



4.1 Oracle® BI Presentation Services Component

To achieve better performance with Presentation Services (OBIPS) component, the following parameters can be tuned in *instanceconfig.xml*, *config.xml* file for better performance and scalability.

Charting threads / memory related tunable parameters

Number of charting threads and maximum number of jobs allowed in the queue can be tuned for performance when the dashboards have several charts:

Important Note: OBIPS has a thread pool for Javahosts. Its size needs to be the total number of threads allowed in all Javahosts. For example, if there are two Javahost instances. Each has 200 threads defined in its own (Javahost) configuration file (*config.xml*), and then the *MaxThreads* size for OBIPS chart threadpool size needs to be 400 (200 + 200).

a. In *instanceconfig.xml* file located at

../user_projects/domains/bi/config/fmwconfig/biconfig/OBIPS,
add the following inside `<ServerInstance>`:

```
<ServerInstance>
  <ThreadPoolDefaults>
    <ChartThreadPool>
      <MinThreads>100</MinThreads>
      <MaxThreads>400</MaxThreads>
      <MaxQueue>2048</MaxQueue>
    </ChartThreadPool>
  </ThreadPoolDefaults>
</ServerInstance>
```

b. In *config.xml* file located at

../user_projects/domains/bi/config/fmwconfig/biconfig/OBIJH,
add the following inside `<JavaHost>`:

```
<JavaHost>
  <JobManager>
    <MinThreads>100</MinThreads>
    <MaxThreads>200</MaxThreads>
    <MaxPendingJobs>200</MaxPendingJobs>
  </JobManager>
</JavaHost>
```

Caching related tunable parameters

There are several OBIPS cache related parameters that can be used to increase OBI PS Caching i.e. number of cache entries, expiry time, and algorithm to clean up the cache etc.

In *instanceconfig.xml* file, add the following inside `<ServerInstance>`:

```
<ServerInstance>
  <Cache>
    <ConnectionPool>
      <MaxAgeMinutes>20</MaxAgeMinutes>
    </ConnectionPool>
    <Query>
      <MaxEntries>5000</MaxEntries>
    </Query>
  </Cache>
</ServerInstance>
```

4.2 Oracle® BI Server Component

The following are the important parameters to tune in the *NQSConfig.INI* file and will increase the performance of the BI system under high users' load:

Initialization blocks

- Repository Init-blocks: Called once during server startup and refreshed as needed after the refresh interval.
- Session Init-blocks: called for each new session; slow SQL can slow down login. Use *Allow deferred execution* option. If you select this option, execution of the initialization block is deferred until an associated session variable is accessed for the first time during the session. **This option prevents execution of all session variable initialization blocks during the session logon stage, giving a shorter logon time.** Session variables that are not needed during the session do not have their initialization blocks executed. This saves the resources which would have been used to execute these unnecessary initialization blocks. See [“When Execution of Session Variable Initialization Blocks Cannot Be Deferred”](#) in Managing Metadata Repositories for Oracle Analytics Server guide.

Number of Init Blocks

For a given session, Init Block queries are executed serially and represent the per session memory costs. Be judicious in creating Init Blocks and disable unwanted init-blocks as slow SQL will slow down login. Verify that the same Init Block query is not already being used in some other Init Block. Verify that cache is enabled for the Init Block and is being utilized.

Parallel Init-blocks execution should be enabled (default serial):

Add below parameter under SECURITY section in NQSConfig.INI:

```
[SECURITY]
```

```
NUM_INIT_BLOCK_THREADS_PER_USER = 4;
```

Init block connection pool and location of data source

As a practice, allocate a separate DB connection pool for Init Blocks [ensure enough connections are available] in the OBIS repository. Init Block query response time will be high if the connection pool points to a remote database.

Init Blocks Cache

- The number of Init Block result sets that are cached with respect to row-wise initialization, can be tuned.
- NQSConfig.INI parameter:

```
[SERVER]
```

```
INIT_BLOCK_CACHE_ENTRIES = 5000
```

BI Database Connection Pool Settings - If you anticipate a higher load on a system, you can change the number of maximum connections for various data sources to make resource use more efficient. The maximum connection size needs to be set in the repository (RPD) for each DB connection pool. The size of the connection pool determines the number of available BI Server connections and the number of available threads for processing physical queries. A logical query may generate multiple physical queries, each of which could go to different connections.

Use OBIS performance counters in EM to determine if more or fewer DB connections are required. *For a simple sizing calculation, let's assume there are peak N users concurrently downloading dashboard pages. On average, each dashboard page executes L logical queries. On average, each logical query executes P physical queries. Then the number of DB connections required for this load would be $N * L * P$.*

Important Note: *If fewer connections are specified, then physical query response times will increase.*

Tuning of OBI Server session and threads

- NQSSConfig.INI parameters to be tuned:

[SERVER]

MAX_SESSION_LIMIT = 2000;

#The above parameters specifies the number of sessions that can be connected to BI Server even if inactive. The sessions and the corresponding queries are queued to the threads for processing as they become available. Typically, the number of sessions specified by **MAX_SESSION_LIMIT** is higher than the number of available threads specified by **SERVER_THREAD_RANGE**. Set the **MAX_SESSION_LIMIT** to a value that reflects the maximum number of users per node + 10% for sessions yet to be timed-out.

SERVER_THREAD_RANGE = 80-1000;

#The above parameter specifies the number of threads that process the logical queries—the number of queries that can be active in the BI Server at any time. When the number of required threads goes beyond 80, threads will be created and destroyed as on a demand basis. For a typical OAS implementation, a setting of 80-1000 is sufficient. This value may be increased if load and stress tests prove more threads are beneficial for the throughput and response time of the BI Server.

DB_GATEWAY_THREAD_RANGE = 80-1000;

#The parameter **DB_GATEWAY_THREAD_RANGE** in the Server section of **NQSSConfig.INI** establishes when Oracle BI Server terminates idle threads. The lower number in the range is the number of threads that are kept open before Oracle BI Server takes action. If the number of open threads exceeds the low point in the range, Oracle BI Server terminates idle threads. For example, if **DB_GATEWAY_THREAD_RANGE** is set to 80-100 and 175 threads are open, Oracle BI Server terminates any idle threads back down to the 80 minimum thread limit.

See "[A NQSCONFIG.INI File Configuration Settings](#)" in Administering Oracle Analytics Server guide.

Aggregate tables: It is extremely important to use aggregate tables to improve query performance. Aggregate tables contain pre-calculated summarizations of data. It is much faster to retrieve an answer from an aggregate table than to re-compute the answer from thousands of rows of detail.

The Oracle BI Server uses aggregate tables automatically, if they've been properly specified in the repository. See *Managing Metadata Repositories Guide* for Oracle Analytics Server for examples of setting up aggregate navigation.

Query Plan Caching - The query plan cache is an internal performance feature that increases the speed of the query compilation process by caching plans for the most recently used queries.

- When the Query Plan cache is hit :
 - It eliminates query parsing time.
 - It increases scalability due to less lock contention.
- Never set the query plan cache size to 0. Doing so may cause Result Cache misses.
- NQSCONFIG.INI parameters to be tuned:

[SERVER]

MAX_QUERY_PLAN_CACHE_ENTRIES = 1024; # default is 1024

MAX_QUERY_PLAN_CACHE_ENTRY_SIZE = 0; # 0 for default

Note: Based on testing you need to put in the right value for your system, see "[NQSCONFIG.INI File Configuration Settings](#)" in Administering Oracle Analytics Server guide for more information about query caching concepts and setup.

Query Results Caching

- One of the main advantages of query caching is the improvement of query performance. It might be valuable to seed the cache during off hours by running queries and hence causing the server to cache their results.
- The number of cache entries and disk size limit for the cache can be configured.
- Note parameters can be managed by either Fusion Middleware Control or by editing NQSCONFIG.INI. NQSCONFIG.INI parameters to be tuned:

[CACHE]

ENABLE = YES;

MAX_ROWS_PER_CACHE_ENTRY = 100000; #Too many rows in cache can slow down performance

#This Configuration setting is managed by Oracle Analytics Server Enterprise Manager

MAX_CACHE_ENTRY_SIZE = 40 MB;

#This Configuration setting is managed by Oracle Analytics Server Enterprise Manager

MAX_CACHE_ENTRIES = 5000;

POPULATE_AGGREGATE_ROLLUP_HITS = YES;

#Above parameter specifies whether to aggregate data from earlier cached query

#results set and create a new entry in the query cache for rollup cache hits. The

#default value is NO. When this parameter is set to YES, then when a query gets an #aggregate rollup hit, the result is put into the cache. Setting this parameter to YES might #result in better performance but results in more entries being added to the cache.

You can configure the Oracle BI Server to maintain a local, disk-based cache of query result sets (query cache). The query cache allows the Oracle BI Server to satisfy many subsequent query requests without having to access back-end data sources (such as Oracle Database). This reduction in communication costs can dramatically decrease query response time. See "[About the Oracle BI Server Query Cache](#)" in the Administering Oracle Analytics Server guide.

Read-Only Mode - Permits or forbids changing Oracle BI repository files when the Administration Tool is in either online or offline mode. The default is NO, meaning that repositories can be edited.

- Makes the repository read-only so that online updates cannot be made.
- Increased scalability due to less lock contention
- NQSCONFIG.INI parameters to be tuned via EM Console:

[SERVER]

This Configuration setting is managed by Oracle Analytics Server Enterprise Manager

READ_ONLY_MODE = YES;

See "[A NQSCONFIG.INI File Configuration Settings](#)" in Administering Oracle Analytics Server guide.

Improve sort efficiency by increasing sort buffer

- It helps to have the sort directory on a fast disk (e.g. a RAMDisk)
- NQSCONFIG.INI parameters to be tuned:

[GENERAL]

WORK_DIRECTORY_PATHS = "C:\Temp"; /* /dev/shm on Linux */

#If a Memory Resident File System is not available, for optimum performance,

#temporary directories should reside on local high performance storage devices (i.e.

#SSD, SAN).

SORT_MEMORY_SIZE = 64 MB;

SORT_BUFFER_INCREMENT_SIZE = 256 KB;

Cluster aware Cache seeding (using nqcmd or Agent)

- Seeding one node propagates across cluster
- Data from shared location is pulled into local cache location during every poll
- NQSCONFIG.INI parameters to be tuned:

[CACHE]

This Configuration setting is managed by Oracle Analytics Server Enterprise Manager

GLOBAL_CACHE_STORAGE_PATH = "<shared directory name>" SIZE;

This Configuration setting is managed by Oracle Analytics Server Enterprise Manager

MAX_GLOBAL_CACHE_ENTRIES = 5000;

```
CACHE_POLL_SECONDS = 300;
```

See "[A NQSConfig.INI File Configuration Settings](#)" in Administering Oracle Analytics Server Guide.

Minimizing Impact on Query Performance Due to Network Latency

Network latency increases when Oracle Analytics Server connects to a database on a different data center. You can minimize the negative performance impact from this higher network latency by reducing the number of network roundtrips between Oracle Analytics Server and your data source. By making fewer network roundtrips, you can reduce the response time of your database queries.

To achieve this, it is recommended that you increase the default bulk fetch row count and the size of the fetch buffer. You can modify both below DB Features settings in the data model file (RPD) for your Oracle Analytics Server.

- **DEFAULT_BULK_FETCH_ROW_COUNT:** Increase the bulk fetch row count value to **5000**. As the number of rows in bulk fetches is limited by the size of the bulk fetch buffer, you need to increase the fetch buffer size too.
- **MAX_BULK_FETCH_BUFFER_SIZE:** Increase the buffer size to a large value, such as **10000000**.

Note: If the buffer size is not high enough, each bulk fetch contains fewer rows than expected by the `DEFAULT_BULK_FETCH_ROW_COUNT`. When network latency is high, this can degrade performance.

Query Optimization Flags Section Parameters

In NQSConfig.INI file the parameters in the Query Optimization Flags section can override the behavior of Oracle BI Server. The Query Optimizer Thread Pool handles the complex queries where the query execution plan (rqTree) is too deep to put into one thread, which causes the stack to overflow. The Query Optimizer Thread Pool uses parameters that enable you to divide the execution plan, letting each thread in that pool handle a part of the plan. For more details on how to tune these parameters, refer to section [Query Optimization Flags Section Parameters](#) in the Administering Oracle Analytics Server guide.

Avoid Excessive BI Logging

Avoid excessive writing to log files in following BI Server components because this can quickly create a disk bottleneck under high users load test.

a. OBIS Query Loglevel:

Set *Loglevel 2* this will provide the logical query, physical query and the response times. Note: It has reasonable low overhead in low to moderately busy system and can be kept at level 2 in production.

b. Set OAS log level to 'Error':

In EM, set default log level for all BI System components to "Error".

4.3 Oracle® JavaHost Component

The following are the important parameters to tune in *JavaHost config.xml* and *obijh.properties* files and will increase the performance for scenarios with Trellis charts rendering and export to excel downloads.

4.3.1 Tuning Exporting Large Data Sets to Microsoft Excel

To improve export to excel download time, it is recommended to perform following fine tuning for the JavaHost process.

Important Note: *To avoid potential system performance issues, the following tuning values for JavaHost parameters should be set only after considering the nature of the Excel download workload along with available system memory capacity.*

Java Virtual Machine Settings

In *obijh.properties* file located at `ORACLE_HOME/bi/modules/oracle.bi.cam.obijh/env/` for JavaHost (64bit) set heap size to *from* `-Xmx1024M` to `-Xmx8g`. In following section of *obijh.properties* file, increase the heap size to 8GB for 64bit JavaHost process:

```
MEM_ARGS=-Xmx8g
```

Set 0 for XMLP tag in JavaHost configuration file to handle large data

Navigate to `ORACLE_HOME/user_projects/domains/bi/config/fmwconfig/biconfig/`, `OBIJH` folder.

In the `config.xml` file, configure the XMLP tag for large data as follows:

```
<XMLP>
  <InputStreamLimitInKB>0</InputStreamLimitInKB>
  <ReadRequestBeforeProcessing>false</ReadRequestBeforeProcessing>
</XMLP>
```

Important Note: Setting `InputStreamLimitInKB` parameter value to zero (0), which is unlimited, should only be used for testing. Set the value to something reasonable that works with your large data sets. The default is 8192 (8MB), but you may need to increase it to 16384 (16MB), 32768 (32MB) etc.

Move temporary files locations for JavaHost / OBIPS to fast storage

At least 4.5GB of free temporary space is required for single user export (126821 rows and 90 columns), multiplied by the number of concurrent users who will export such large reports at the same time. Note: the temp file size will be improved in future BI patch sets.

For **JavaHost** the default location for temporary files location is in `../bi/system_components/OBIJH/obijh1/tmp` and for **OBIPS** the default location for temporary folder is in `../user_projects/domains/bi/servers/obips1/tmp` sub-folders.

Important Note: As temporary files locations are highly used by JavaHost / OBIPS processes, any improvement on IO performance (i.e. using Flash drives, RAMDISK) for these tmp files locations would significantly improve performance of the export to excel.

On Linux you can use RAMDISK to put tmp files for JavaHost and OBIPS processes. **Caution: You need at least 256GB free physical memory in the system in order to implement following RAMDISK tuning settings. Please note the settings below will cause severe performance issues if implemented in system that has low memory.**

- a. For JavaHost, you need to add `-Djava.io.tmpdir=/dev/shm` argument in `obijh.properties` file located at `ORACLE_HOME/bi/modules/oracle.bi.cam.obijh/ervl/` , in following section of `obijh.properties` file:

`OBIJH_ARGS=-server -Djava.io.tmpdir=/dev/shm/jh1`

- b. For OBIPS, you need to perform following to move `tmp` files location to RAMDISK:

```
mkdir -p /dev/shm/obips1/tmp
cd ../user_projects/servers/obips1
mv tmp obips1_tmp.localDisk
ln -s /dev/shm/obips1/tmp
```

Note: make sure these paths, permissions and directory stays persist after reboot. Consult your Linux documentation for more information on how to implement this.

4.4 Oracle® BI Schedulers Component

To achieve better performance with the BI Schedulers component, the following parameters can be tuned in `schedulerconfig.xml`

You may encounter the following behavior with large Agent runs. A description of the issue and a possible tuning solution follows:

- For personalized Agents with large email distributions that include PDF attachments, some concurrency issues between OBIPS and OBISched may be encountered. These will typically manifest as missed email deliveries and corrupt PDF attachments. To alleviate such occurrences, the OBIScheduler parameter '`MaxExecThreads`' should be relatively less than the OBIPS parameter '`MaxConcurrentRequests`'.

Note: The setting in `instanceconfig.xml` (`MaxConcurrentRequests`) affects only PDFs by throttling their generation.

The setting in `scheduler` (`MaxExecThreads`) affects all delivers agents by throttling OBISched requests to OBIPS." In the installed default configuration, these values are set as follows:

OBIPS '`MaxConcurrentRequests`' = 100

OBISched '`MaxExecThreads`' = 200

Important Note: These settings may produce issues for the situation detailed above as OBISched could overwhelm OBIPS with requests. As mentioned above, '`MaxExecThreads`' should be relatively less (25% to 50%) than '`MaxConcurrentRequests`'.

Following settings are recommended as a starting point,

- a. In OBIPS `instanceconfig.xml` file located at

`/user_projects/domains/bi/config/fmwconfig/biconfig/OBIPS,`

add the following inside `<ServerInstance>`:

```
<Download>
  <XslFo>
    <MaxConcurrentRequests>100</MaxConcurrentRequests>
  </XslFo>
</Download>
```

- b. In OBISched `schedulerconfig.xml` located at

`/user_projects/domains/bi/config/fmwconfig/biconfig/OBISCH,`

update the following parameter value inside `<ServerInstance>`:

```
<MaxExecThreads>200</MaxExecThreads>
```

Best Practices Guide for Tuning Oracle® Analytics Server

Applies To: Oracle Analytics Server (5.x and later)

First edition: May 2022

Author: Ahmed Awan

Oracle Analytics Service Excellence (OASE) - Customer Engineering & Advocacy Lab - CEAL Team

Change Record

Date	Primary Author	Version	Change Reference
052022	Ahmed Awan	1.0	First edition

Oracle® is a registered trademark of Oracle® Corporation. Various product and service names referenced herein may be trademarks of Oracle® Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2022 Oracle® Corporation
All rights reserved.