Deep Dive into Oracle Access Management 12.2.1.4.0 Performance on Oracle Container Engine for Kubernetes

An Oracle White Paper, 1st Edition

June 8, 2021 | Version 1.0

TABLE OF CONTENTS

PURPOSE STATEMENT

This document discusses performance recommendations and sizing for Oracle Access Manager (OAM) as part of Oracle Identity Management (IDM) Suite Release 12.2.1.4.0 on Oracle Container Engine for Kubernetes (OKE).

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

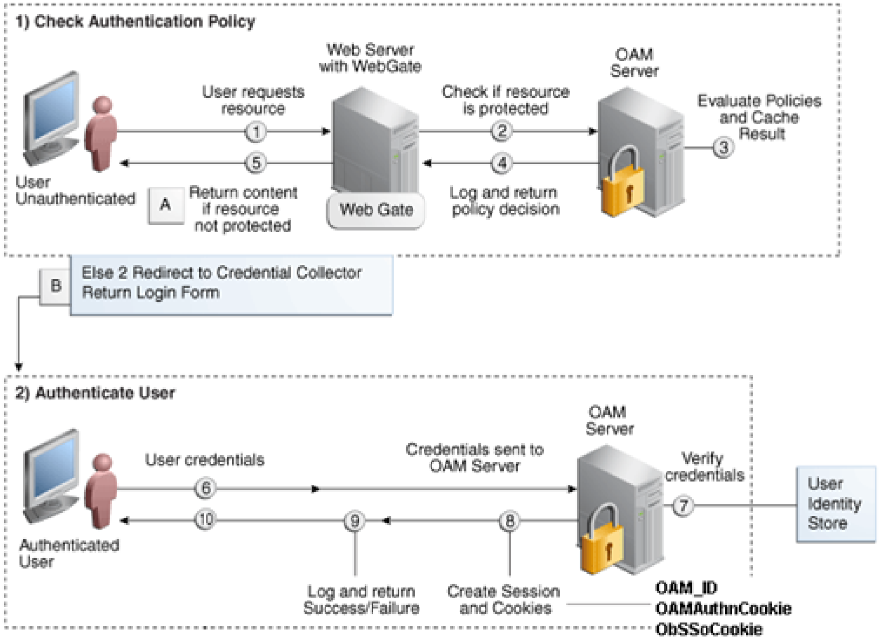
INTRODUCTION

With increasing requirements for high scalability and performance in the field of Identity Management,
Oracle has conducted scalability benchmarks for Oracle Access Manager (OAM) as part of Oracle Identity Management (IDM) Release 12.2.1.4.0, 2021 12c PS4 April BP. The environments for OAM scalability benchmarks were deployed on Oracle Container Engine for Kubernetes (OKE) and DBaaS shapes in Oracle Cloud Infrastructure (OCI). The objective was to measure OAM user authentication flow scalability for up to 80 thousand (80K) users, seeded in Oracle Unified Directory (OUD), measure the load for OAM cluster running up to 20 OAM OKE PODs with four OAM PODs per Node, provide the tuning recommendations and project the scalability for higher number of OAM users.

ORACLE ACCESS MANAGER TEST CASE OVERVIEW

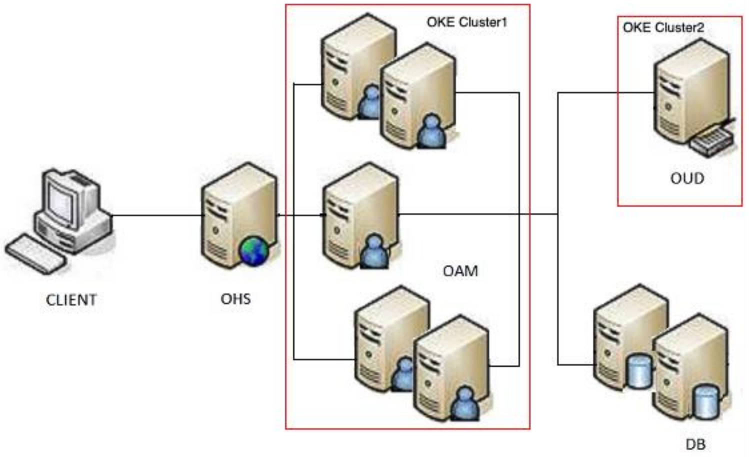
Oracle Access Manager offers access control services to provide centralized authentication, policy-based authorizations, and auditing. To measure the performance and scalability, several OAM authentication tests were executed to show OAM scaling up on a single node and scaling out for two nodes in OCI virtual compute shapes.

The OAM authentication test case involves a user navigating to a website URL protected by OAM. The WebGate plug-in in the Web Server will intercept the request and check with the OAM server to see if the user has been authenticated and has a valid session. If not, the OAM Server will redirect the user to an OAM login page where a username and password can be submitted. The OAM server then evaluates the authentication policy and authenticates against the OUD directory. Once the user is authenticated, a user session is created, and authorization policies are checked to see if the user is allowed access to the protected resource. If authorized, the user will be redirected to the requested page. OAM 12c uses database for server-side session management. In other words, session data is stored (and persisted) on the database server side. OAM sessions represent state that is associated with the user’s login and resource access, which is utilized to manage the user’s access to OAM protected resources. Some of the information that is managed includes authorization and authentication events during user access. The users are defined under a single hierarchical group, which gives access to the protected resources by simple LDAP authentication. Refer to the sequence diagram below, depicting the described flow.


1) Check Authentication Policy
Web Server with WebGate
OAM Server
User requests resource
Check if resource is protected
Evaluate Policies and Cache Result
1
2
3
4
5
User Unauthenticated
A
Return content if resource not protected
Web Gate
Log and return policy decision
B
Else 2 Redirect to Credential Collector Return Login Form
2) Authenticate User
OAM Server
User credentials
Credentials sent to OAM Server
Verify credentials
User Identity Store
6
7
8
9
10
Authenticated User
Log and return Success/Failure
Create Session and Cookies
OAM_ID
OAMAuthnCookie
ObSSoCookie

Topology

Refer to the schematic picture below, describing the test configuration with all required OAM tiers, collocated in the same network segment to eliminate network latencies in the scalability runs.



Hardware Configuration

Depending on the number of concurrent users OAM configurations can be categorized as small, medium and large sized implementations. This section covers recommended hardware shapes, offered in Oracle Cloud Infrastructure (OCI) for each size to accommodate for the concurrent load and projects for the future growth. The OAM customers can use the OCI shape specifications as the guidance for deploying comparable on-premise OAM configurations. Refer to the following link for more details on OCI compute shapes: https://docs.oracle.com/en-us/iaas/Content/Compute/References/computeshapes.htm

Table 1-1: OAM Compute NODE recommended configurations

Configuration	Small	Medium	Large
Concurrent OAM Users	**Max 16,000**	**16,000 to 32,000**	**32,000 and higher**
OCI shape for OAM Node	Standard2.4	Standard2.8	Standard2.16
OCI shape RAM, Gb	60	120	240
OCI shape OCPUs	4	8	16

Table 1-2: OAM Oracle RDBMS configuration

Configuration	Small	Medium	Large
OAM RAC DB Node	Standard2.4	Standard2.4 - 2.8	Standard2.8 - 2.16
OCI shape RAM, Gb	60	60 - 120	120 - 240
OCI shape OCPUs	4	4 – 8	8 - 16

Table 1-3: OUD Compute NODE configuration

Configuration	Small	Medium	Large
OCI shape for OUD Node	Standard2.4	Standard2.4 - 2.8	Standard2.4 - 2.16

OCI shape RAM, Gb	60	60 - 120	60 – 120
OCI shape OCPUs	4	4 – 8	4 – 16

Table 1-4: OHS compute node configuration

Configuration	Small	Medium	Large
OHS Compute	Standard2.4	Standard2.4 - 2.8	Standard2.8 - 2.16
OCI shape RAM, Gb	60	60 - 120	120 - 240
OCI shape OCPUs	4	4 – 8	8 - 16

Important!

- OAM compute sizing specifications apply to a single NODE, hosting up to four OAM Docker PODs. You can deploy another POD (up to 4) or another Node, using the same shape / specifications to scale-out and run as an OAM cluster.
- OAM database can handle user authentication load from more than one OAM node, so the recommended specifications apply to a two node RAC database instance. You will have to closely monitor the database load when you add more OAM Compute nodes and enable more concurrent users in your environments. Use of Oracle Database Real Application Cluster (RAC) is recommended to provide better load balancing and OAM database scale-out for large configurations.
- OUD compute with single POD on single Node can sustain higher concurrent load, so you can use smaller shapes for Small and Medium configurations. Large configuration extends to high-end workload, so it has the wider range for its compute specifications.
- OHS compute VM can handle the load for more than one OAM Node. You can consider deploying more than one OHS VM to ensure load balancing in OAM environments.

OAM AUTHENTICATION BENCHMARK RESULTS

This chapter provides the results of OAM load using authentication scenario, described earlier, for Small, Medium and Large sample configurations. OUD has been seeded with 200K users for all configurations.

‘Small’ Configuration Benchmarks

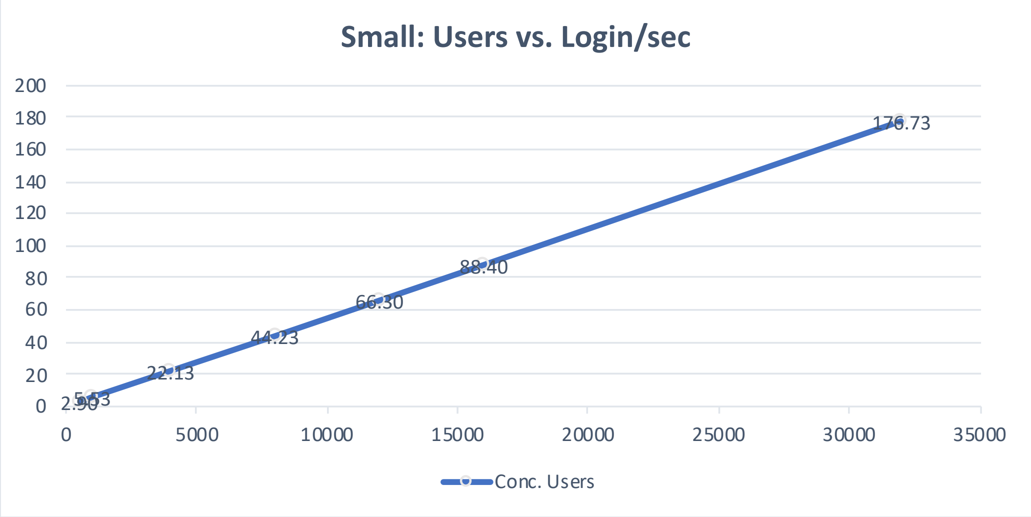
The ‘Small’ configuration used Standard2.4 shapes for all its tiers, OAM, database, OUD and OHS.

Table 2-1: Small Configuration Benchmarks

Conc. Users	Login / hour	Login / sec	Avg RT, sec	90% RT, sec	OHS CPU Usage %	OAM DB1 CPU Usage %	OAM DB2 CPU Usage %	OUD CPU Usage %	OAM1 CPU Usage %	OAM2 CPU Usage %	OAM3 CPU Usage %	OAM4 CPU Usage %	OAM5 CPU Usage %	OAM6 CPU Usage %	OAM7 CPU Usage %	OAM8 CPU Usage %
500	10.4K	2.90	0.021	0.025	0.21	5.05	4.49	3.70	15.07							
1000	19.9K	5.53	0.023	0.026	0.24	5.31	4.95	2.51	16.24							
4000	79.7K	22.13	0.025	0.027	2.88	6.75	6.66	5.54	48.41							
8000	159.2K	44.23	0.036	0.045	3.04	8.96	9.96	9.40	58.02	60.58						
12000	238.7K	66.30	0.037	0.041	5.73	11.35	12.41	12.79	60.88	61.30	65.58					
16000	318.2K	88.40	0.049	0.068	6.47	16.68	15.28	18.33	67.47	68.67	70.39	71.06				
32000	636.2K	176.73	0.059	0.094	13.57	24.11	33.75	33.98	75.03	74.59	73.23	73.10	68.61	71.44	71.03	69.30

Note: OAM and OUD CPU% values above are for POD level usage.

Figure 2-1: Number of Users to TPS for OAM ‘Small’ configurations



The following are the observations from the user authentication scalability benchmarks for up to eight-node OAM cluster for ‘Small’ configuration:

- OAM has demonstrated near-linear scalability for the number of transactions per second (TPS) with the number of users for 4 PODs for one NODE and eight PODs for two NODEs.
- The average response time has been consistent in the range of 0.02 seconds for a single NODE with four OAM PODs but went up to 0.059 sec for two NODEs with a total of eight PODs (4 PODs per NODE) when the load doubled in the system.
- The CPU% utilization of OAM POD reached ~70%, on single NODE w/ 4 OAM PODs for the workload of 16000 concurrent users, and on two NODEs running w/ 8 OAM PODs(4 OAM POD per NODE) for the workload of 32,000 concurrent users.
- The OUD POD CPU% utilization reached ~18% for single NODE w/ 4 OAM PODs w/ 16,000 concurrent users, and ~33% for two NODEs w/ 4 OAM PODs each w/ 32,000 concurrent users.
- OAM two RAC node DB CPU utilization has reached ~24% and ~33% per instance when the 2nd OAM NODE was added to the system and sustained the load for ‘Small’ configuration.
- OHS has not manifested any noticeable resource spikes and handled the system load for all tried scenarios.

‘Medium’ Configuration Benchmarks

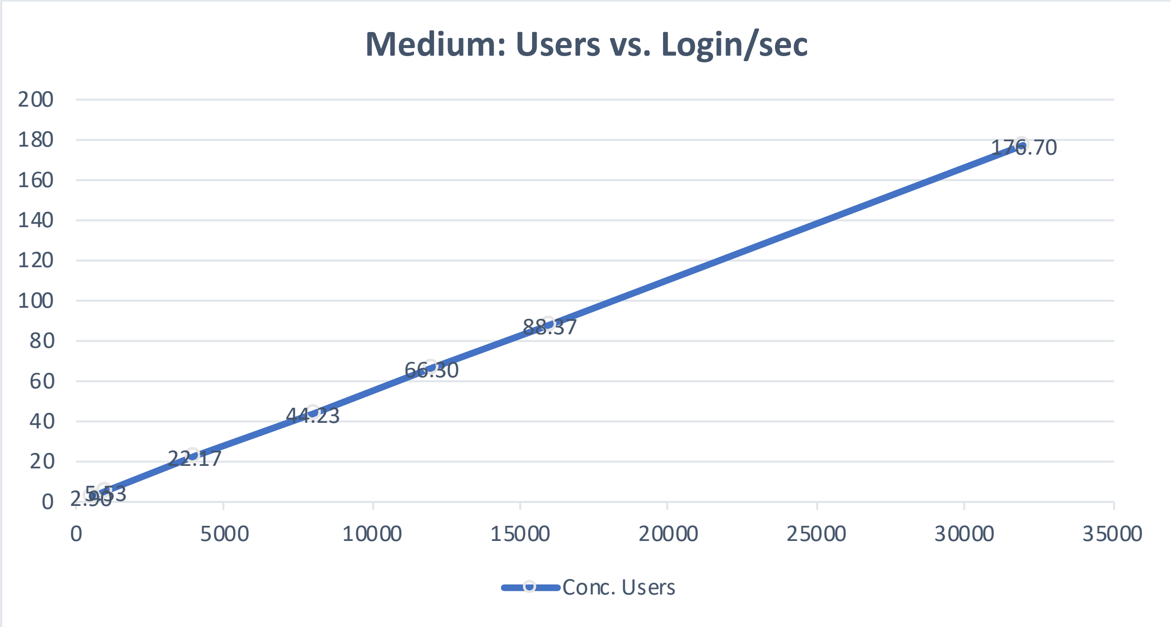
The ‘Medium’ configuration used Standard2.8 shapes for all its tiers, OAM, database, OUD and OHS.

Table 2-2: Medium Configuration Benchmarks

Conc. Users	Login / hour	Login / sec	Avg RT, sec	90% RT, sec	OHS CPU Usage %	OAM DB1 CPU Usage %	OAM DB2 CPU Usage %	OUD CPU Usage %	OAM1 CPU Usage %	OAM2 CPU Usage %	OAM3 CPU Usage %	OAM4 CPU Usage %	OAM5 CPU Usage %	OAM6 CPU Usage %	OAM7 CPU Usage %	OAM8 CPU Usage %
500	10.4K	2.90	0.021	0.025	0.08	2.41	2.42	2.51	14.88							
1000	19.9K	5.53	0.023	0.026	0.09	2.44	2.21	2.91	15.67							
4000	79.8K	22.17	0.024	0.027	1.04	3.18	2.95	5.54	48.41							
8000	159.2K	44.23	0.033	0.036	1.09	5.20	4.61	9.24	55.53	55.43						
12000	238.7K	66.30	0.034	0.036	2.06	7.18	6.52	11.73	59.76	58.33	57.88					
16000	318.2K	88.37	0.042	0.044	2.33	9.70	9.46	15.32	64.17	62.83	61.68	63.65				
32000	636.2K	176.70	0.045	0.047	4.88	14.13	13.03	30.33	67.27	62.66	65.01	65.06	67.44	64.82	66.11	65.72

Note: In the above table OAM and OUD CPU% is for POD level CPU Usage.

Figure 2-2: No. of Users to TPS for OAM ‘Medium’ configurations



The following are the observations from the user authentication scalability benchmarks for up to eight-node OAM cluster for ‘Medium’ configuration:

- OAM has demonstrated near-linear scalability for the number of transactions per second (TPS) with the number of users for four PODs for one and eight PODs for two OAM NODEs.
- The average response time has been consistent in the range of 0.02 seconds for a single NODE with four OAM PODs but went up to 0.045 sec for two NODEs with a total of eight PODs (4 PODs per NODE) when the load doubled in the system.
- The CPU% utilization of OAM POD reached ~65%, on single NODE w/ 4 OAM PODs for the workload of 16000 concurrent users, and on two NODEs running w/ 8 OAM PODs(4 OAM POD per NODE) for the workload of 32,000 concurrent users.
- The OUD POD CPU% utilization reached ~15% for single NODE w/ 4 OAM PODs w/ 16,000 concurrent users, and ~30% for two NODEs w/ 4 OAM PODs each w/ 32,000 concurrent users.
- OAM two RAC node DB CPU utilization has reached ~13-14% per instance when the 2nd OAM NODE was added to the system and sustained the load for ‘Medium’ configuration.
- OHS has not manifested any noticeable resource spikes and handled the system load for all tried scenarios

‘Large’ Configuration Benchmarks

The ‘Large’ configuration used Standard2.16 shapes for all its tiers, OAM, database, OUD and OHS.

Table 2-3: Large Configuration Benchmarks

Conc. Users	Login / hour	Login / sec	Avg RT, sec	90% RT, sec	OHS CPU Usage %	OAM DB1 CPU Usage %	OAM DB2 CPU Usage %	OUD1 CPU Usage %	OUD2 CPU Usage %	OAM1 CPU Usage %	OAM2 CPU Usage %	OAM3 CPU Usage %	OAM4 CPU Usage %	OAM5 CPU Usage %	OAM6 CPU Usage %	OAM7 CPU Usage %	OAM8 CPU Usage %
500	10.2K	2.83	0.019	0.023	0.03	1.18	1.06	1.15		7.94							
1000	19.9K	5.53	0.020	0.024	0.80	1.24	1.08	2.14		14.47							
4000	79.7K	22.13	0.022	0.025	0.83	1.44	1.33	4.31		42.62							
8000	159.8K	44.40	0.026	0.029	0.98	1.96	1.84	7.75		51.66	49.24						
12000	238.7K	66.30	0.026	0.025	1.01	2.44	2.40	11.85		51.98	50.47	50.74					
16000	318.2K	88.40	0.032	0.035	1.02	3.18	2.87	14.68		56.60	56.06	53.95	56.30				
32000	635.4K	176.50	0.036	0.044	2.02	7.26	6.01	26.87		59.47	61.58	58.09	64.67	52.78	54.17	55.43	53.50
48000	954.4K	265.10	0.040	0.048	3.03	11.30	10.07	39.92		58.83	59.66	55.25	62.81	53.52	51.90	52.29	52.66
64000	1271.9K	353.30	0.045	0.056	4.06	12.43	15.47	52.54		61.33	61.01	63.01	62.81	54.99	54.59	57.10	56.14
80000	1589.5K	441.53	0.057	0.070	5.08	20.76	18.73	42.10	33.59	64.76	64.74	60.89	63.81	56.98	52.88	56.44	56.04

Table 2-3: Large Configuration Benchmarks (Con’d)

Conc. Users	OAM9 CPU Usage %	OAM10 CPU Usage %	OAM11 CPU Usage %	OAM12 CPU Usage %	OAM13 CPU Usage %	OAM14 CPU Usage %	OAM15 CPU Usage %	OAM16 CPU Usage %	OAM17 CPU Usage %	OAM18 CPU Usage %	OAM19 CPU Usage %	OAM20 CPU Usage %
500												
1000												
4000												
8000												
12000												
16000												

32000												
48000	56.37	56.24	55.65	53.76								
64000	53.71	55.09	52.41	56.04	53.06	55.09	54.10	53.95				
80000	57.70	56.14	54.55	52.80	53.59	51.78	55.15	51.95	54.20	55.39	52.37	53.90

The following are the observations from the user authentication scalability benchmarks for up to 20-node OAM cluster for ‘Large configuration:

- OAM has demonstrated near-linear scalability for the number of transactions per second (TPS) for all the covered scale-out scenarios.
- The average response time has been consistent in the range of 0.02 seconds for a single NODE with four OAM PODs, and went up to 0.057 sec for five NODEs with 20 OAM PODs.
- The CPU % utilization of OAM POD ranged between 53-64% across all the OAM PODs added to the system and the workload increased to 80,000 concurrent users.
- The CPU % utilization of OUD POD reached ~52% for a single OUD POD and 4 NODEs with 16 OAM PODs, and ~33% and ~42% for two OUD PODs when the 5th NODE with total 20 OAM PODs was added to the system and the workload increased to 80,000 concurrent users.
- OAM two RAC node DB CPU utilization has reached ~21% and ~19% per instance for 5 OAM NODE and sustained the load for ‘Large’ configuration.
- OHS has not manifested any noticeable resource spikes and handled the system load for all tried scenarios
- With the increased load the CPU utilization for the ingress-nginx-controller POD crossed 100%. Creating 10 replicas (2 each on 5 available OAM NODE) for 80,000 user's loads addressed the performance overhead.

Figure 2-3: No. of Users to TPS for OAM ‘Large’ configurations

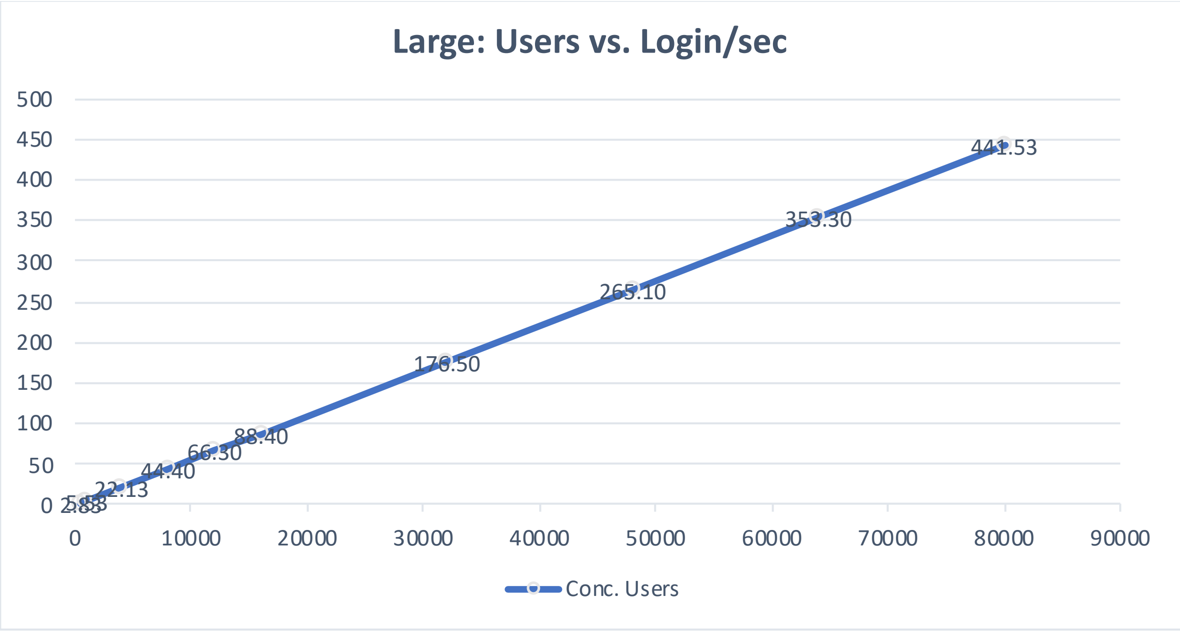
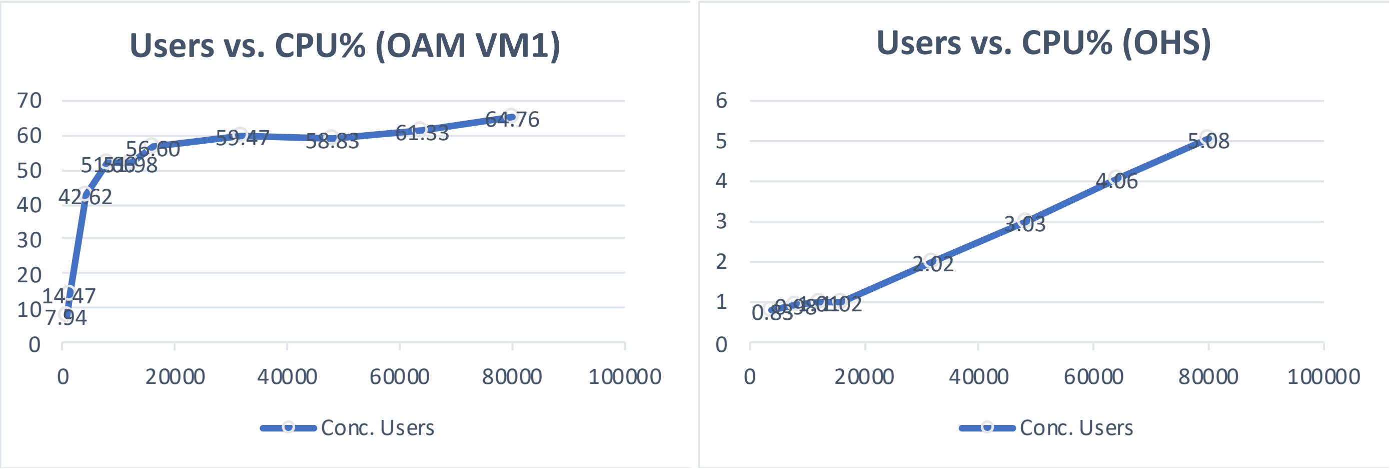
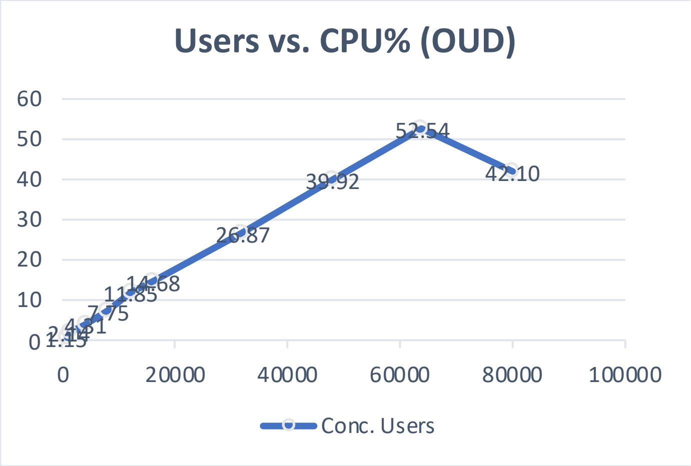


Figure 2-4: CPU% Utilization to No. of Users for a single OAM 'Large' configuration





Note: Drop in the CPU% is due to another OUD POD added and the load distributed across two nodes.

PERFORMANCE TUNING

The performance tuning recommendations are common for all configurations. Shape specific recommendations are referenced explicitly throughout the section.

Webgate /Agent Side Performance Tuning

Webserver Tuning:

1. Set Max Connection to 50 for all managed OAM servers: OAM Console → Application Security → Agents → Server list → Primary Server List → Max Connections =50.
2. Validate the session properties: OAM Console → Configuration → Settings → Common Settings → Session Lifetime (minutes) = 5, Idle Timeout = 15, Max. session per user = 0.

Webgate Tuning:

1. Set Max Connections = 350 (For this sizing activity we tried with this settings)

OUD Tuning:

1. Set your ‘User Identify Store’ Connection pool size: OAM Console → Configuration → User Identity Stores → Select your created Identity store and click Edit → Connection Details → Min and Max Pool Size = 350.

OAM server-Side Tuning

Host Tuning

Update the configuration file /etc/security/limits.conf:

```
*          hard    nofile      500000
*          soft    nofile      500000
root       hard    nofile      500000
root       soft    nofile      500000
*          hard    nproc       500000
*          soft    nproc       500000
```

Weblogic Tuning

OAM JVM Tuning

Apply the below tunings for all OAM servers,

```
JAVA_PROPERTIES="${JAVA_PROPERTIES} -server -Xms8192m -Xmx8192m -XX:+UseParallelGC -XX:+AggressiveHeap -XX:+DisableExplicitGC -
DWLSAGENT_DISABLED=true -XX:ReservedCodeCacheSize=1024m -XX:+PrintGCDetails -XX:+PrintGCDateStamps -
Xloggc:${DOMAIN_HOME}/tmp/GC_${SERVER_NAME}.log -Djava.security.egd=file:/dev/./urandom -Dweblogic.ProductionModeEnabled=true -
Dweblogic.Chunksize=8192 -DMaxRandomPoolSize=1000 -DMaxCipherPoolSize=1000 -DMinRandomPoolSize=1000 -Dweblogic.threadpool.MinPoolSize=300 -
Dweblogic.utils.io.chunkpoolsize=8192 -Doam.smedb.optimizedSessionUpdatesIntervalInMillis=18000 -Doam.smedb.optimizeSessionUpdates=true -
Doracle.oam.SessionStoreStatusCheckMillis=20000"
export JAVA_PROPERTIES
```

OAMDS DataSource Tuning

Update oamDS connection pool settings.

- Login to the WebLogic Server Console, click ‘Lock & Edit’.
- Navigate to Domain Structure → Expand Data Sources → Data Sources → oamDS → Settings for oamDS → Configuration → Connection Pool.
- Update the values for Initial Capacity, Minimum Capacity to 100 and Maximum Capacity to 800
- Click OK and ‘Activate Changes’.

Work Manager Tuning

OAPOverRestWM tuning

Add Minimum Thread constraint to worker manager “OAPOverRestWM”

- Login to the WebLogic Server Console, click ‘Lock & Edit’.
- Navigate to Domain Structure → Deployments → Expand ‘oam_server’ → click click /iam/access/binding → Configuration → Workload → wm/OAPOverRestWM
- Click ‘New’ under ‘Application Scoped Work Managed Components’, select ‘Minimum Threads Constraint’ under ‘Create a New Work Manager Component’, click ‘Next’.
- Set Count=400 under ‘Minimum Threads Constraint Properties’ and click Finish.
- Update the ‘Path’ In ‘Save Deployment Plan’ to the default path of your Plan.xml.
- Click OK and ‘Activate Changes’.

Max Thread/CapacityConstraint Tuning

Remove Max Thread Constraint and Capacity Constraint:

- Repeat the same navigation steps as above.
- Under ‘Application Scoped Work Managed Components’ select the check box for Capacity and MaxThreadsCount. Click Delete.
- In the ‘Delete Work Manage Components’ screen, click OK to delete.
- Click on ‘Release Configuration’ and then Log Out.

OHS Server Tuning

Connect to your OHS Server take a backup and edit its configuration file:
file $ORACLE_HOME/user_projects/domains/base_domain/config/fmwconfig/components/OHS/instances/ohs1/httpd.conf:

```
MaxKeepAliveRequests 0
Timeout 300
KeepAliveTimeout 10
```

Under <IfModule mpm_worker_module>:

```
StartServers              2
ThreadLimit             250
MaxClients             1500
MinSpareThreads         200
MaxSpareThreads         200
ThreadsPerChild         250
MaxRequestWorkers       400
MaxConnectionsPerChild    0
MaxRequestsPerChild       0
Mutex fcntl:${ORACLE_INSTANCE}/servers/${COMPONENT_NAME}/logs
```

Database Tuning

The database memory settings for pga_aggregate_target and sga_target should be set not to exceed 60% of available physical RAM on OAM DB tier.

Refer to the configuration parameters below:

```
fast_start_mttr_target = 300
session_cached_cursors = 1500
sessions               = 4832
Processes              = 3200
audit trail            = DB,EXTENDED
nls_sort               = BINARY
db_securefile          = ALWAYS
plsql_code_type        = NATIVE SCOPE
"_b_tree_bitmap_plans" = FALSE
“_active_session_legacy_behavior" = TRUE
“_optimizer_batch_table_access_by_rowid" = FALSE
pga_aggregate_target  = 7.125G for Small, 14.624G for Medium and 29.625G for Large configuration
sga_target            = 28.5G for Small, 58.5G for Medium and 118.5G for Large configuration
```

CONCLUSION

This document consolidates the best practices and recommendations for sizing Oracle Access Manager 12.2.1.4.0 to ensure best performance and scalability.

GLOSSARY

- OKE - Container Engine for Kubernetes
- IDM – Oracle Identity Management
- OAM – Oracle Access Manager
- OCI – Oracle Cloud Infrastructure
- OHS – Oracle HTTP Server
- OUD – Oracle Unified Directory
- RAC – Oracle Real Application Cluster
- RT – Response Time
- TPS – transactions per sec

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Deep Dive into Oracle Access Management 12.2.1.4.0 Performance on Oracle Container Engine for Kubernetes
May 11, 2021
Contributing Authors: Muthukumar Karunanidhi, Pavel Buynitsky, Vijay Mangalgi,