ORACLE

Automatic Scaling of Oracle Autonomous Database with Oracle JD Edwards EnterpriseOne

Performance Characterization of Automatic Scaling of Oracle Autonomous Database with Oracle JD Edwards EnterpriseOne

August 2023 | Version 2.1

Purpose Statement

The Automatic Scaling capability of the Oracle Autonomous Database (ADB) architecture provides an automatic way to adjust the number of compute resources (OCPUs) in an instance pool. Auto scaling provides a consistent performance solution for EnterpriseOne software peak workloads that require high database resources and helps reduce the overall costs during times of average workloads.

This document describes the results of testing the EnterpriseOne software in an Oracle Autonomous Database architecture, to provide a clear picture of the behavior of auto scaling. This is aimed at providing a level of assurance to customers adopting EnterpriseOne with Oracle Autonomous Database at its Automatic scaling capabilities.

Disclaimer

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.

Table of Contents

List of Figures

List of Tables

Executive Summary

Oracle Autonomous Database is the world's first autonomous data management in the cloud to deliver automated patching, upgrades, and tuning, which includes performing all routine database maintenance tasks while the system is running without any human intervention. Autonomous Database is self-driving, self-securing, and self-repairing, which helps to eliminate manual database management and human errors.

One of the core capabilities of Oracle Autonomous Database is automatic scaling of database resources which makes it fully elastic. Customers simply specify the number of OCPUs and the storage capacity for the database, and at any time, based on the changes in the workload, the database resources will automatically increase or decrease to provide optimal performance. With auto scaling enabled, the database can use up to three times more CPU and IO resources than those that are associated with the specified number of OCPUs. The scaling of OCPU resource can also be done manually.

The automatic scaling capability of Oracle Autonomous Database eliminates the manual overhead of database management and resource allocation for Oracle database customers. Auto scaling also ensures optimal utilization of database resources without the cost associated with an over-provisioned architecture. Automatic scaling ensures that the required level of database resources is made available for the peak resource-intensive workloads that EnterpriseOne customer might require, which ensures optimal level of cost to performance. This provides overall significant cost savings and improved productivity to EnterpriseOne customers.

EnterpriseOne Performance Benefits Through Autonomous Database Using Auto Scaling

The performance characterization of EnterpriseOne software with Oracle auto scaling enablement provides a profile and the behavior of EnterpriseOne software with this Oracle functionality.

Table 1 illustrates the test cases and corresponding performance improvements of JD Edwards EnterpriseOne interactive applications, batch processes, and orchestrations when auto scaling is enabled in the Oracle Autonomous Database.

Test Case	Performance Benefit - Autoscaling Enabled
Interactive Processes	60%
Batch Processes	70%
Orchestrations	67%

Table 1. Performance Benefit of Auto Scaling Enabled

As shown in the test scenarios and their corresponding performance benefits in Table 1, a minimum of 60% improvement was observed in all the workloads.

Overview

Auto scaling is one of the key features of an Oracle Autonomous Database. Auto scaling is enabled by default during the provisioning of the Autonomous Database. Auto scaling feature allows the EnterpriseOne workload to use three-times the available CPU processing power and I/O bandwidth resources. Auto scaling aids in providing optimal performance for resource-intensive workloads in a cost-efficient manner, as it eliminates the need for over-provisioning database resources and the costs associated with it.

The Oracle Autonomous Database architecture is provisioned with a base number of OCPUs, and the pricing structure starts with the cost for those base OCPUs. For billing purposes, whether auto scaling is enabled or not, a customer will always be charged for the provisioned base number of OCPUs. Auto scaling enablement charges for any additional OCPUs utilized for only the period of time that exceeds the base allocation of OCPUs.

The performance testing of Oracle Autonomous Database auto scaling with EnterpriseOne involves a typical customer profile performing standard activities such as interactive user load, batch processing, EnterpriseOne orchestrations, and EnterpriseOne Search processing. Appendix A describes typical customer load scenarios that form the basis of comparison for the Oracle auto scaling configuration.

Oracle Autonomous Database auto scaling enablement measures the impact on the EnterpriseOne application testing performance. The auto scaling feature uses the Autonomous Database resource manager to control the number of OCPUs and I/O bandwidth available to EnterpriseOne. When auto scaling is disabled, the Autonomous Database resource manager uses the base provisioned number of OCPUs and I/O resources for the database. In this document, this value is 4 OCPUs. When auto scaling is enabled, the Autonomous Database resource manager is adjusted to allow up to a three-fold increase or 12 OCPUs and a three-fold increase in I/O bandwidth.

Enablement of auto scaling in an Autonomous Database provides more database resources to process the load during higher than base allocated OCPU traffic needs. As a business perfective, enabling auto scaling is a good approach to ensure smooth and consistent customer business process behavior. The additional charges associated with the Oracle pay-per-use model for enabling auto scaling come into play when OCPU resource consumption exceeds the base OCPU allocation.

Customers can also manually scale up their database resources. Manual scaling is the user defined OCPU count value for the Oracle Autonomous Database. By default, 4 OCPUs are provisioned as the base number of OCPUs at the ADBS build time irrespective of whether auto scaling is configured. The user can modify the OCPU count value at any time through the Oracle cloud dashboard.

Manual scaling is used to modify the base number of OCPUs and is used to change how the customers pay-per-use starting cost value is calculated. For example, if a customer provisioned the Autonomous Database with 4 OCPUs and later decided to increase the OCPU count to 6 manually then the customer has to increase the count from 4 to 6 OCPUs through the manage resource allocation ADBS cloud dashboard interface. Similarly, if the customer wants to decrease the OCPUs count value from 6 OCPUs to 4 OCPUs it can be achieved through the same ADBS cloud dashboard interface.

Appendix B covers the setting of the ADBS base number of OCPUs and configuring the Oracle Autonomous Database for manual scaling. The main goal of this document is to demonstrate Oracle Autonomous Database auto scaling enablement in an EnterpriseOne software environment under typical load conditions.

Test Methodology and Metrics

The performance characteristics of the EnterpriseOne applications configured with Oracle Autonomous Database and its auto scaling functionality are measured by running the workloads that can stress the EnterpriseOne server components. The workload was run in EnterpriseOne with and without the enablement of Oracle Autonomous Database auto scaling feature. The methodology section covers the tests involved and categorizes the test processes.

Methodology

Table 2 illustrates the tests that are run against the EnterpriseOne environment and test processes that are involved in the testing. The EnterpriseOne test processes run concurrently, keeping the Autonomous Database auto scaling feature enabled. An identical test has been run in the same database with auto scaling feature disabled.

TEST CASE	TESTCASE PROCESSES
1. Auto Scaling Enabled	- Interactive User - Batch - Orchestration - EnterpriseOne Search
2. Auto Scaling Disabled	

Table 2. Testcase and Testcase processes

Interactive user load and EnterpriseOne Search are processed through the EnterpriseOne web interface, batch processes are initiated on the EnterpriseOne Enterprise Server, and EnterpriseOne orchestrations is submitted using EnterpriseOne API calls to the EnterpriseOne AIS server. The detailed explanation of each EnterpriseOne test scenario is discussed in Appendix A under the EnterpriseOne Load Test Cases section.

Metrics

Metrics indicate if the application is performing within the expected boundaries or whether it is performing below acceptable standards. CPU utilization and process runtimes metrics help in the quantification of important data on workloads run against the EnterpriseOne environment. Selected Oracle Cloud Database metrics collected during the load test and corresponding runtimes averages of interactive and batch processes are discussed in the following test results section.

Test Results

The results of EnterpriseOne runtime metrics and database graph metric are divided into two discussions, one with the auto scaling feature enabled and the other with the auto scaling feature disabled. The analysis of metrics such as CPU utilization, database thread utilization, interactive/batch runtime results, and average active session (AAS) measurements is discussed.

CPU Utilization

The Oracle DB CPU measurement is the amount of CPU time (microseconds) consumed on requests made at the database level. In terms of CPU utilization, the load test performed in EnterpriseOne with auto scaling feature enabled in Autonomous Database demonstrated a significant improvement over the same test with auto scaling disabled.

Figure 1, which is a CPU consumption graph taken from the Oracle Autonomous Database cloud console under database monitoring tools, shows the percent CPU utilized throughout the load test submitted to EnterpriseOne environment configured with Oracle Autonomous Database and auto scaling feature enabled.

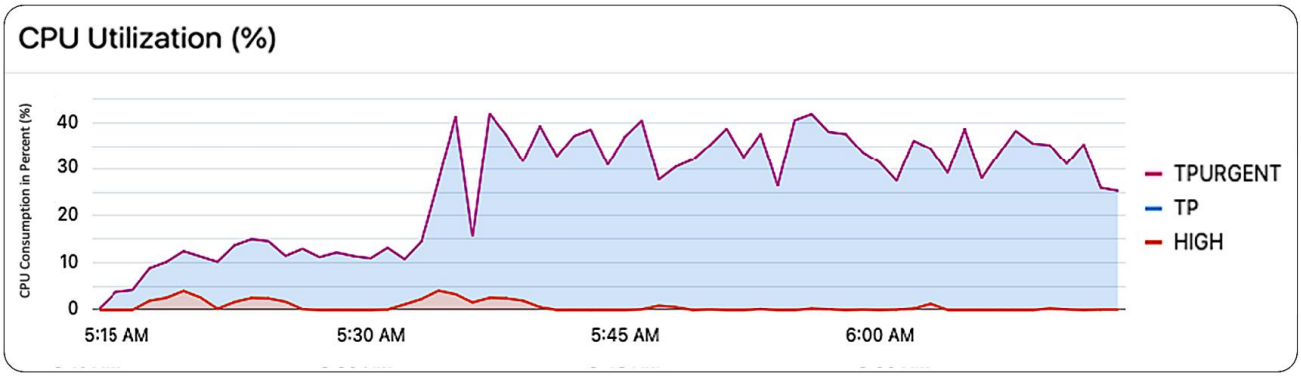

Figure 1. CPU Consumption – Auto Scaling Enabled

The processes in Figure 1, representing interactive, batch, orchestrations, and EnterpriseOne search running concurrently, observed an average of 30% CPU utilization at the steady state achieved just after 5:30 AM with auto scaling enabled.

Three services are measured in Figure 1: TPURGENT, TP and HIGH. TPURGENT represents the highest priority application connection service and HIGH represents high priority application connection services respectively. The CPU utilization of the Transaction Processing (TP) service is the area hightligthed under the TPURGENT line. Transcation processing is the key indicator for measuring performance for EnterpriseOne covered in this discussion.

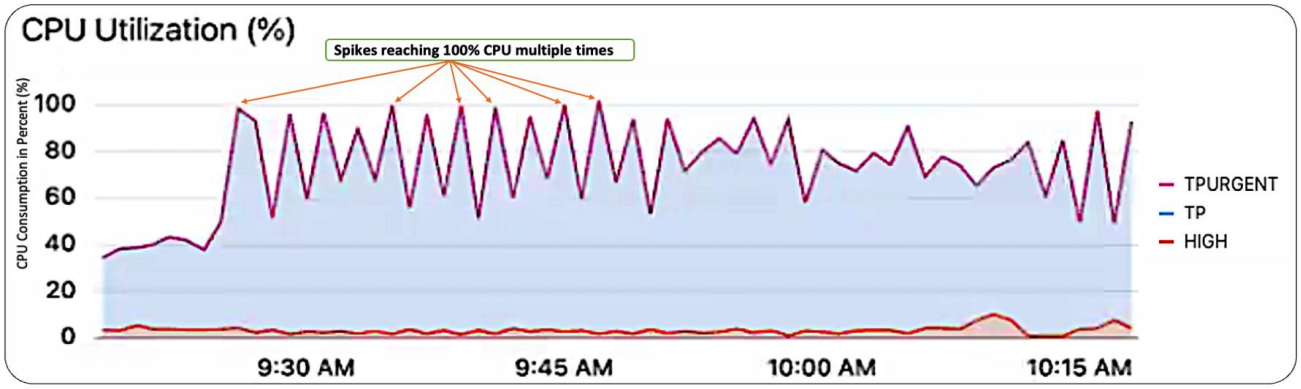

Figure 2. CPU Consumption – Auto Scaling Disabled.

In contrast, an identical test is conducted, as shown in Figure 2, where auto scaling is disabled. In Figure 2, CPU utilization reaches the 100% mark several times, measuring an overall average of 90% CPU utilization. At the moment where the CPU reaches 100%, EnterpriseOne processes will queue up in a wait state and this will lead to an increase in response time for the application. The correlating metrics measuring the performance of EnterpriseOne is presented later in this document in the discussions around Figures 7 and 8.

Database Thread Utilization

Oracle database objects that process program instructions within the Oracle database are called threads. Database threads usage is a crucial measure for showing how well Oracle Autonomous Database performs when auto scaling is enabled.

In this scenario, an Autonomous Database with 4 OCPUs and the auto scale feature enabled can scale up to 12 OCPUs or 24 threads, whereas an Autonomous Database with same 4 OCPUs and the auto scale feature disabled can only use up to 4 OCPUs or 8 threads (that is, 2 threads per OCPU).

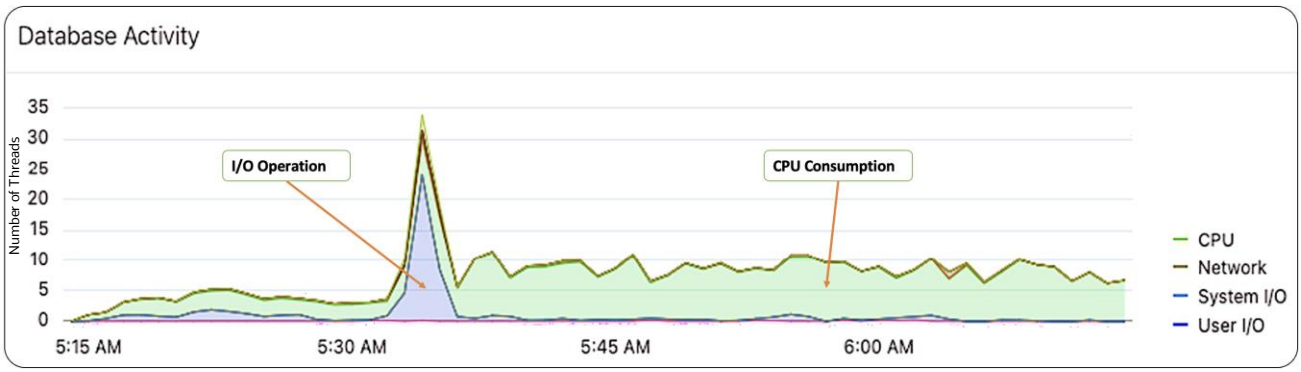

Figure 3. ADB Thread Utilizations - Auto Scaling Enabled

The Database activity, as shown in Figure 3, utilized 10 threads of CPU and IO resources on average in a steady-state load test with auto scaling feature enabled. The shaded area pointed to the CPU consumption indicates the threads utilized by the CPU, and the shaded area pointed to the I/O represents threads used by IO operations.

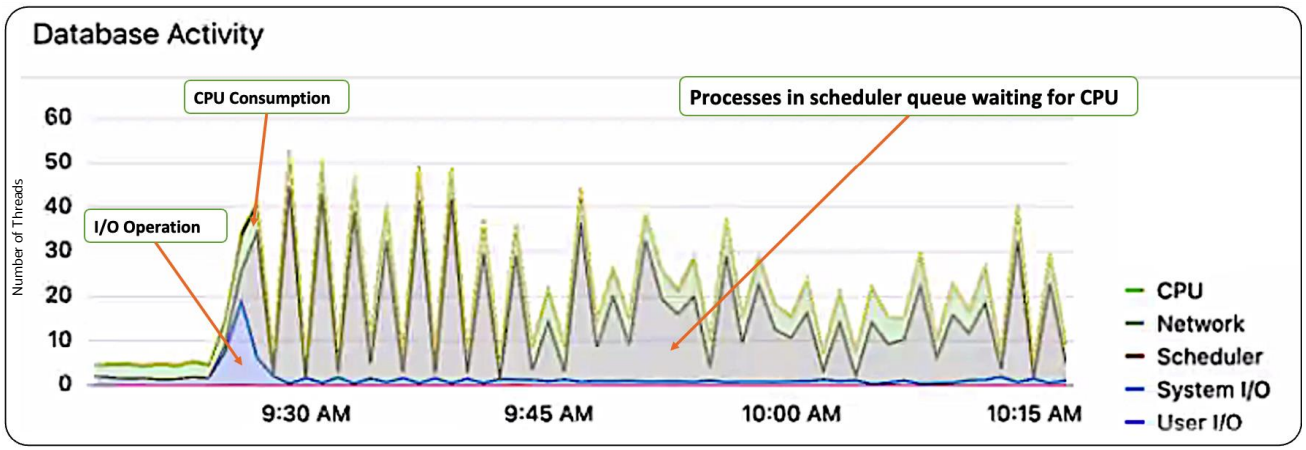

Figure 4. ADB Thread Utilizations - Auto Scaling Disabled

Figure 4 depicts the database activity of the same load test workload but with auto scaling disabled. The shaded region where the scheduler queue has large spikes indicates that a large amount of the processes are waiting for CPU, which further correlates to the 100% CPU utilization spikes as shown in Figure 2.

The database has a maximum limit of 8 threads with auto scaling disabled. Application threads that exceed this limit will have to wait for CPU resources until the active threads are complete. Threads in the wait queue signify that there are insufficient resources and the processes using those threads will then take a longer time to complete creating a performance degradation situation.

Database Activity Summary

In summary, Figure 3 has the following characteristics with auto scaling enabled:

- Provides the 3 fold increase, changing the available threads from 8 to 24.
- 10 of the 24 total threads provided by auto scaling enabled managed the demand created by EnterpriseOne application.
- 14 of the 24 threads were unused and are readily available for futher application demands.

In contrast, as illustrated in Figure 4, in the scenario where auto scaling is disabled, the number of available threads exceeds. This creates a performance bottleneck due to the lack of available threads to service EnterpriseOne application requests resulting in processes in the scheduler queue waiting for CPU resources.

Average Active Sessions

Average Active Sessions (AAS) is the ratio of the change in database time (time in the database in seconds spent in CPU, IO and waiting states) divided by the clock time. AAS metric is often used as an indicator of database activity to provide a sense of how many sessions the database utilizes to perform the workload provided to the database. At any one time, the AAS measurement reflects the number of sessions that are either functioning or waiting for a resource.

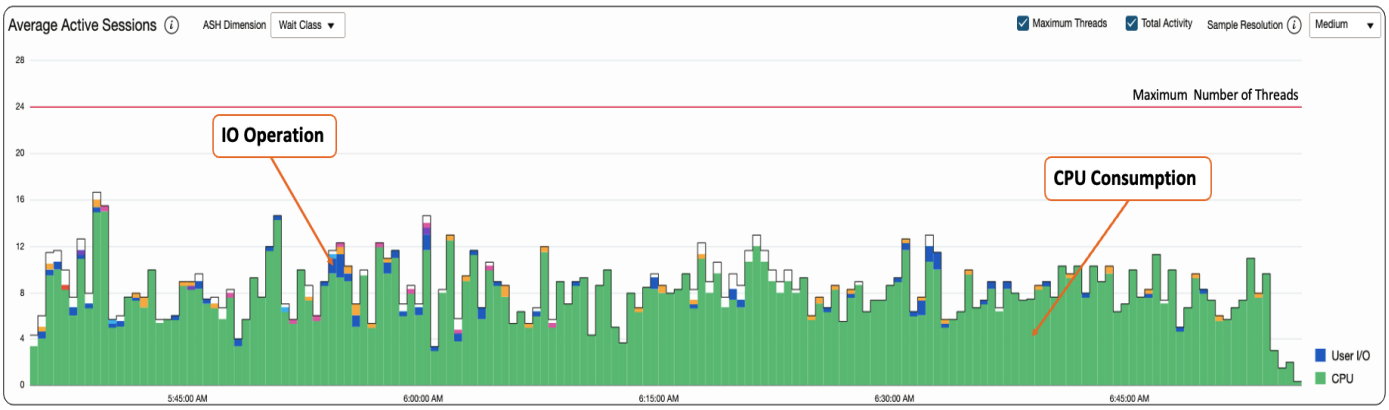

Figure 5. Average Active Sessions – Auto Scaling Enabled

Figure 5 depicts the database average active sessions that are using CPU and IO resources. The figure points to different areas indicating either IO operations and CPU consumption. The horizontal line at 24 average active sessions represents the maximum number of threads available to the EnterpriseOne application. Figure 5 illustrates the demand delivered to the database by EnterpriseOne applications uses 8 to 10 threads on average out of a total of 24 available threads. Figure 5, where auto scaling is enabled, shows a profile where there is no performance impact related to CPU resource constraints.

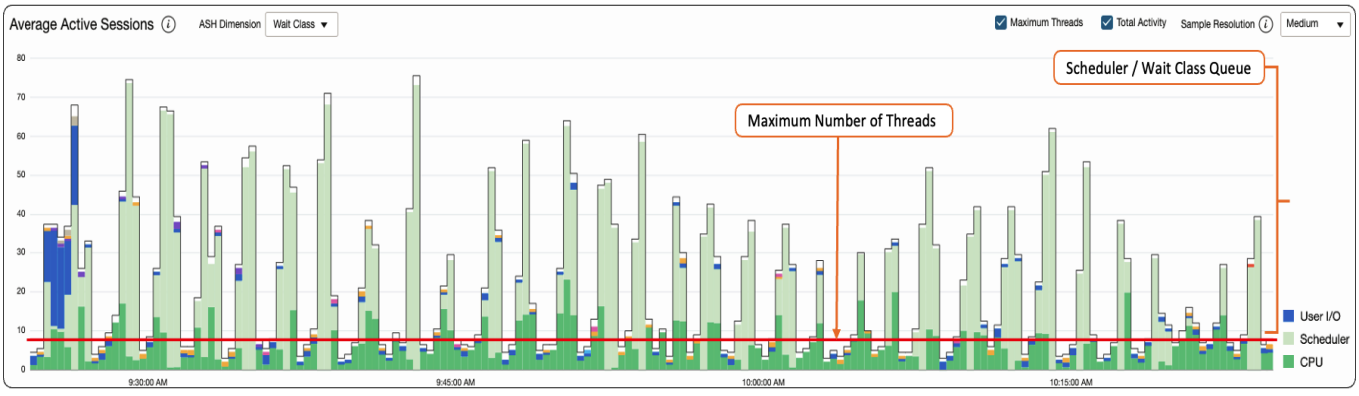

Figure 6. Average Active Sessions – Auto Scaling Disabled

Figure 6, on the other hand, shows the average active sessions with auto scaling disabled. The region above the horizontal line at 8 threads indicates the processes exceeding the maximum number of threads allowed by the Autonomous Database. The region above the horizontal line indicates programs that are in a wait/scheduler queue

waiting for CPU resources, which represents a profile that indicates a possible performance impact to the EnterpriseOne application.

Active Sessions Analysis Summary

As seen in Figure 5, when auto scaling is enabled, none of the processes enter a wait state because sufficient resources are available to meet the requirements of the workload. Whereas in Figure 6, auto scaling is disabled, threads are shown in a wait state, impacting the EnterpriseOne test load.

Application Timing Results

Application timing results measured by processing interactive, batch, orchestration, and EnterpriseOne search are discussed in this section. The overall analysis of the results show a performance improvement in the runtimes of all the EnterpriseOne application processes where auto scaling is enabled.

Interactive Runtimes

Runtime averages of interactive user processes are presented in this section. Test scenario simulating a series of web interface application interactions exercising specific EnterpriseOne modules, including actions such as logging in to application and performing a sequence of specific module flows typically used by a customer in their day-to-day activities.

Figure 7 illustrates the runtime results of the interactive modules. In all test cases a performance benefit is observed when auto scaling is enabled. Table 3 in Appendix A outlines the interactive processes and the corresponding modules the applications belong to.

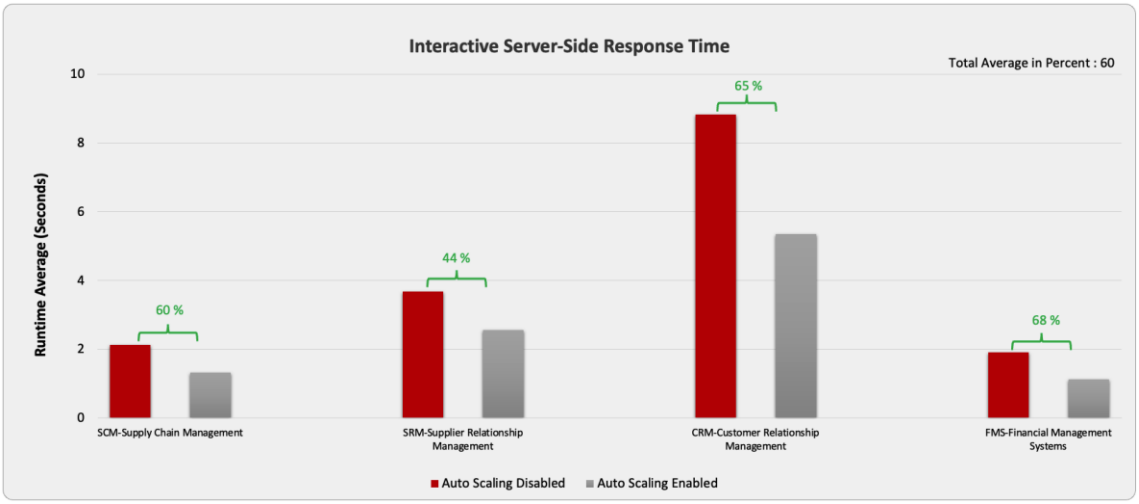

Figure 7. Interactive Module Averages

Three of the four modules showed a 60% or more improvement with one module performing only 44% better. Overall, EnterpriseOne interactive testing averaged 60% performance improvement in average runtime with auto scaling enabled.

Batch Runtimes

A batch process is defined as a non-interactive method of submitting a request to the EnterpriseOne application. Batch process runtimes is the metric used for EnterpriseOne auto scaling enablement comparison in this section. Batch processes can generate higher load than interactive user processes and thus are impacted to a greater extent by CPU and IO resource limitations.

Figure 8 shows the batch performance improvement comparison with auto scaling feature enablement.
The objective of this section is to illustrate the advantages of auto scaling and its three-fold increase in CPU and IO resources.

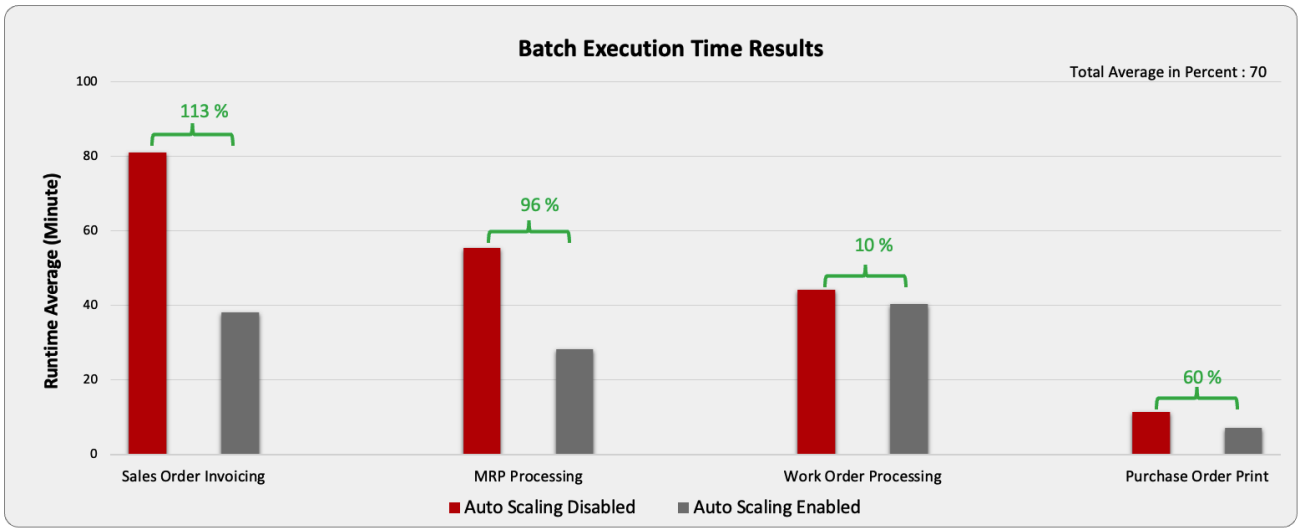

Figure 8. Batch Runtime Percent Variance in Minutes

Figure 8 illustrates the batch performance improvement with auto scaling enabled, where three of the four batch processes averaging a 90% or better, and one of the processes having a performance of 10% improvement. Overall, with auto scaling enabled in the database, batch performance improved by 70%.

Orchestration Results

EnterpriseOne orchestrations enhance the transaction-based EnterpriseOne system of records allowing a system to also provide a dynamic representation of a customer's real-time business operations.
Orchestrations are run on the EnterpriseOne AIS server and used here to produce additional load for auto scaling enablement comparison.

Figure 9. Orchestrations Runtime in Seconds

Orchestrations selected for the load testing are presented above in the above figure. Detailed explanations of each Orchestration are provided in Appendix A.

An overall benefit of 67% was observed with the load when auto scaling is enabled. This further shows an improvement in performance with an EnterpriseOne process as a result of increased availability of CPU and IO resources that comes with enabling auto scaling.

Conclusion

The EnterpriseOne software deployment with Oracle Autonomous Database configuration and its automatic scaling feature significantly improved the performance of all the EnterpriseOne applications tested, including interactive, batch, orchestration, and EnterpriseOne Search.

Enablement of auto scaling in the database eliminates the need for human intervention to manually adjust CPU and IO resources due to an increase in EnterpriseOne application needs. As a result, while processing the resource intrinsic workloads, constant database availability and day-to-day business operations are achieved. Thus, auto scaling provides a customer an opportunity to increase productivity, save resources, and reduce significant cost, which in turn is used to drive innovation in critical areas of their organizations.

Appendix A: EnterpriseOne Testing Architecture

The EnterpriseOne testing architecture is a hierarchical flow of how load is processed by the different EnterpriseOne server components used to test the Oracle Autonomous database auto scaling enablement feature. Figure 11 illustrates the components of EnterpriseOne technology layer and how they are tied to the Autonomous Database.

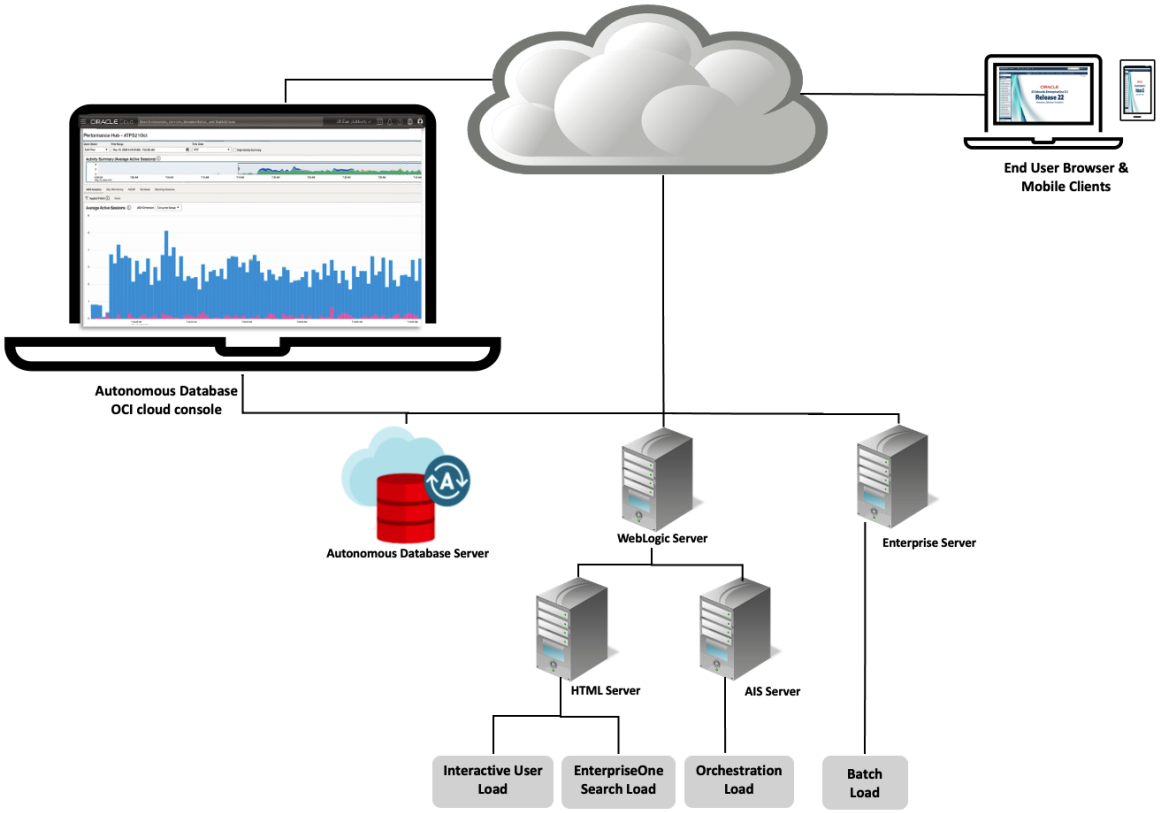

Figure 10. EnterpriseOne environment architecture

As illustrated in the above Figure, the EnterpriseOne server component that is used in the testing architecture includes the HTML Server, which processes the interactive user and EnterpriseOne Search workloads. EnterpriseOne batch workload is processed through the Enterprise Server component and EnterpriseOne orchestration workload is processed through EnterpriseOne AIS Server.

Test Processes

EnterpriseOne workloads are divided and distributed across four test processes in order to effectively test the auto scaling capability of the Oracle Autonomous Database from the varied EnterpriseOne architecture components. Figure 12 details the specific processes that consists the varied EnterpriseOne workloads.

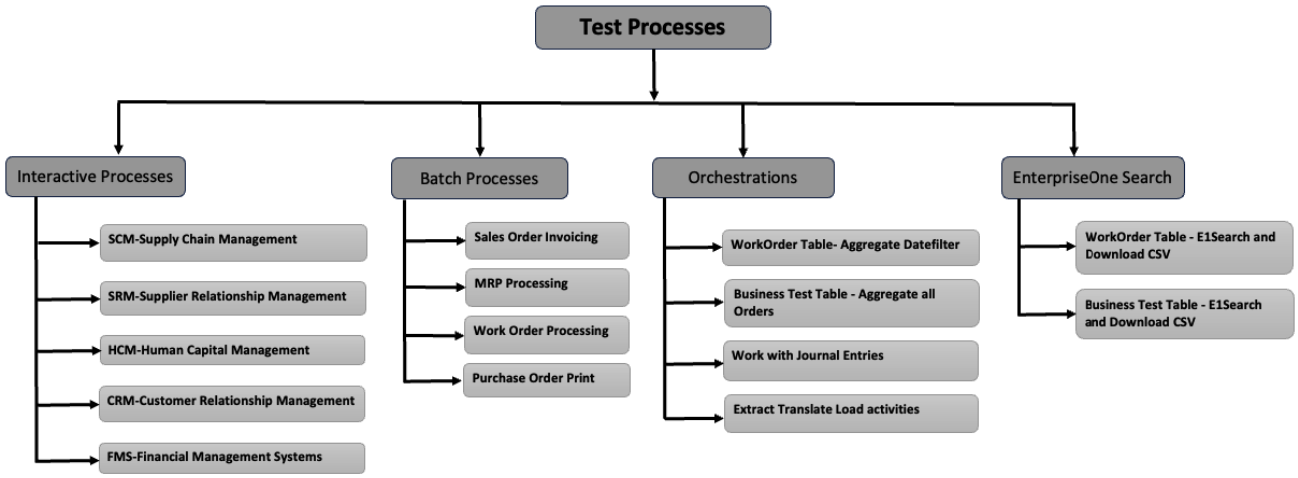

Figure 11. Test Processes and Workload

The **Interactive processes** represent five of the major modules in JD Edwards EnterpriseOne including sales, procurement, human resources, manufacturing, and finance.

Interactive Module Name	Application Name
Customer Relationship Management	Case Management Add
Financial Management Systems	Apply Accounts Receivable Receipt
	Supplier Ledger Inquiry
Human Capital Management	Employee Daily Time Entry
Supplier Relationship Management	Purchase Order Entry
	Purchase Order Receipt
	Voucher Match
Supply Chain Management	Sales Order Entry
	Sales Order Update
	Shipment Approval
	Shipment Confirmation
	Shipment Change and Confirmation
	Inventory Transfer
	Work Order Completion
	MRP Messages for Work Orders
	MRP Messages for Purchase Orders
	MRP Messages for Transfer Orders

Table 3. Interactive processes

A **Batch process** or Universal Batch Engine (UBE) is a non-interactive process workload that is also represented by the sales order invoicing, MRP processing, work order processing, purchase order print.

The **EnterpriseOne Orchestration** workload generates the real time dynamic business activities specific to EnterpriseOne modules. These included the work order table - aggregate date filter, business test Table - aggregate all Orders, work with journal entries, extract translate load activities.

The **EnterpriseOne Search** provides business view information across one or more EnterpriseOne tables for easier navigation through EnterpriseOne application.

Similar to interactive workload, EnterpriseOne Search is processed through the HTML Server and includes a work order and business test table search. There were no metrics collected or presented in this document for EnterpriseOne Search workloads. The primary goal of including EnterpriseOne Search as a workload was to produce greater load to the Autonomous Database for testing.

Appendix B: Understand Oracle ADB Auto Scaling

Auto scaling is a feature in the Oracle Autonomous Database that allows the database to utilize up to three-times the amount of CPU and IO resources than the standard base allocated resources.

Figure 14, is an interface in the Autonomous Database Cloud Console to manage the setting of the base number of OCPUs count value. Oracle Autonomous Database auto scaling feature can be enabled by selecting the OCPU auto scaling check box as shown in the following figure. The selected checkbox shows the enablement of the Oracle Autonomous auto scaling feature. Select this option for a three-fold increase of CPU and IO resources . At a detail level, on the Autonomous Database Server, selecting auto scaling adjusts the resource manager to increase the allowed CPU and IO resources available to the environment.

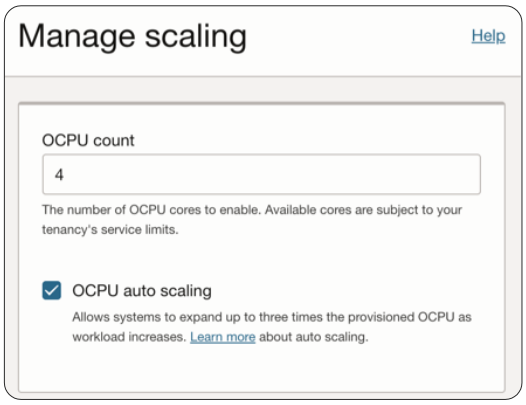

Figure 12. Manage scaling in OCI console

The Autonomous Database auto scaling model is based on a pay-per-use approach. If the customer utilizes additional OCPUs, then customers will incur an extra charge for the duration of time the resources are used.
Auto scaling has the added benefit of the additional three-times increase in IO resources in the form of larger IOPS and throughput, regardless of whether they exceed OCPU base limit. The pay-per-use model is limited to additional OCPU usage only for increased costs incurred and is not dependent on any IO usage. Therefore, with auto scaling enabled, the customer can benefit immediately from an increase in IO resources at no expense, until the workload for OCPU exceeds the base or original OCPU count.

Manual Scaling: Manual scaling is setting the base OCPU count value for the Autonomous Database. The count value is modified in the OCPU count field provided in the Manage scaling window as shown in the following figure, where the base OCPU count value is 4. If the OCPU count value is changed and applied, the Autonomous Database can take up to 5 minutes to adjust the resource manager to the new value.

Connect with us

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

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