A TO Z OF ORACLE ON AZURE

Table of Contents

Building Oracle on Azure IaaS

Oracle is Much More than Just Relational Workloads

Apps, database, hardware, virtualization, and cloud

The complexity, size, and high IO demands of Oracle often pose challenges when transitioning to the cloud. Although the primary focus of this document is on the database aspect, it's important to recognize that Oracle operates within a multi-tier system, encompassing web code, applications, hardware, and more.

Throughout the course of this paper, we will consistently refer to Oracle databases as "workloads." Our research indicates that concentrating on the Oracle workload, rather than solely on the database, leads to significantly improved outcomes. Azure provides an avenue for migrating Oracle workloads to align with on-premises setups through an Infrastructure as a Service (IaaS) model. Furthermore, it enables leveraging this critical data for a variety of purposes, including future analytics, data lakes, global data governance, machine learning, and artificial intelligence.

Navigating the intricacies of Oracle can make initiating a cloud migration project seem like a daunting task. The objective of this document is to deconstruct every component related to the database workload tier. It aims to provide a significant starting point and a series of steps that will guide us and our clients towards accomplishing our objectives.

Don't Lift and Shift the Hardware

Over-provisioned

Oracle does not appear to make it easy to migrate anywhere but Oracle Cloud, (OCI):

Penalizing virtualized CPUs.

*"**Microsoft Azure** – count two vCPUs as equivalent to one Oracle Processor license if multithreading of processor cores is enabled, and one vCPU as equivalent to one Oracle Processor license if multi-threading of processor cores is not enabled."*

The 2:1 penalty should not discourage customers migrating to the Azure cloud. We've demonstrated how on-premises database hosts are typically oversized during capacity planning, following a discernible pattern. This over-provisioning is often driven by several factors:

On-premises hardware necessitates generous sizing and padding to meet resource demands over several years, unlike the cloud's ability to scale on CPU. Consequently, on-premises hosts are often purchased larger than required at the time.

Database administrators (DBAs) are instructed to size on-premises hardware to support databases for 2-7 years, relying on capacity growth values and assumptions to project future resource needs.

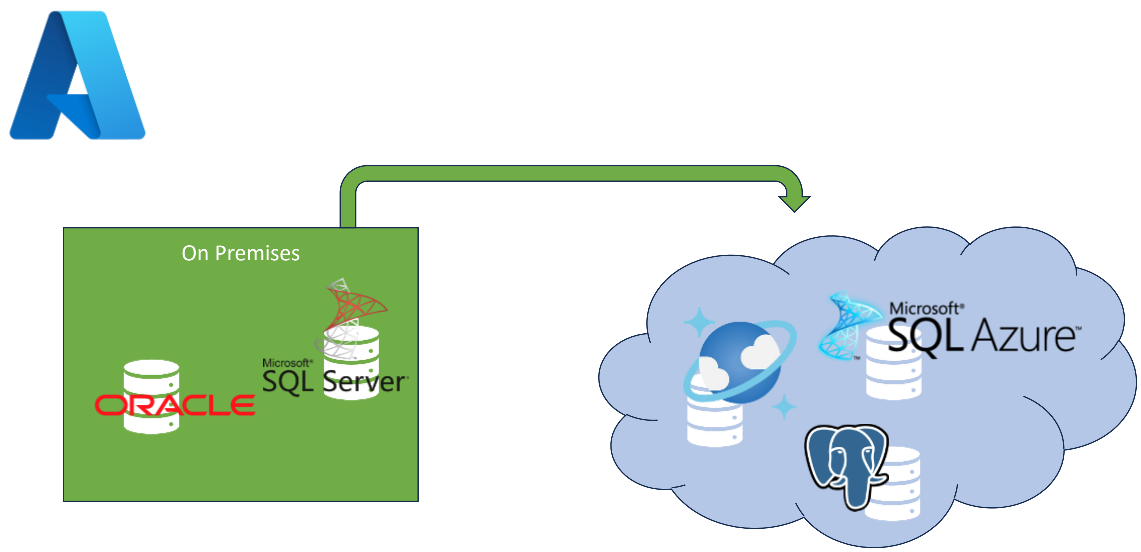
Due to budget considerations, DBAs anticipate that hardware refreshes might not be approved, forcing them to run databases on the original hardware for longer durations. This anticipation leads DBAs to inflate their initial resource estimates.

Workload characteristics have evolved, transforming transactional systems into hybrid environments with increased IO workloads. Improved CPUs have also contributed to better performance, reducing the demand for frequent upgrades.

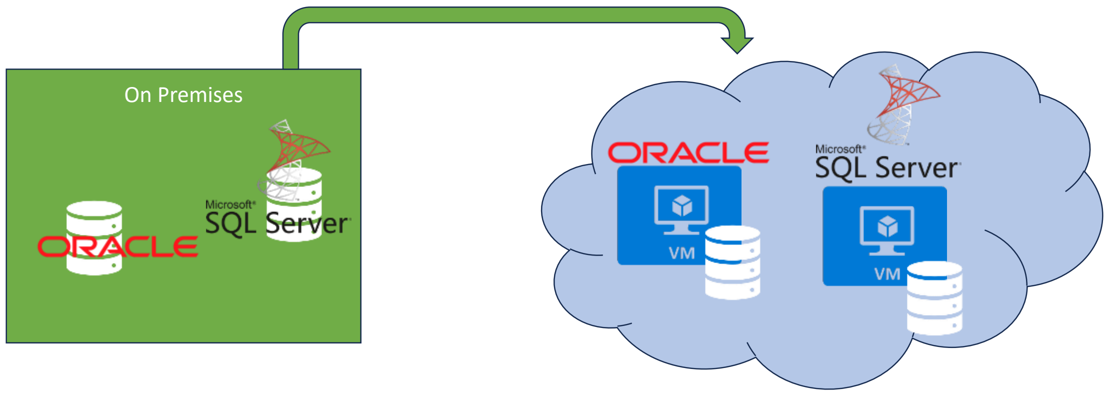
Taking these points into account, it's been established that approximately 85% of assessed Oracle workloads among Azure customers would require only a fraction of the vCPU allocation compared to on-premises systems. The Automatic Workload Repository (AWR) excels in identifying robust workloads, and through a worksheet capable of factoring in averages and aggregated values, it offers accurate estimates for sizing workloads in the Azure cloud.

Oracle for the Lift, Shift and Evolve

Oracle is most often a more difficult refactoring option, but to lift and shift first to the cloud and then to refactor slowly to other options, to an open-source solution, etc., has the most success. This tactic allows to give same features and similar performance as on-premises but allows use of cloud services as part of the migration. It provides extensive time- which is always in short supply, to refactor these massive workloads in the cloud.



For most customers, the migration of the largest and most mission critical workloads often end up requiring unique solutions using Infrastructure as a Service (IaaS).



Don’t Forget the Application and Middleware Tier

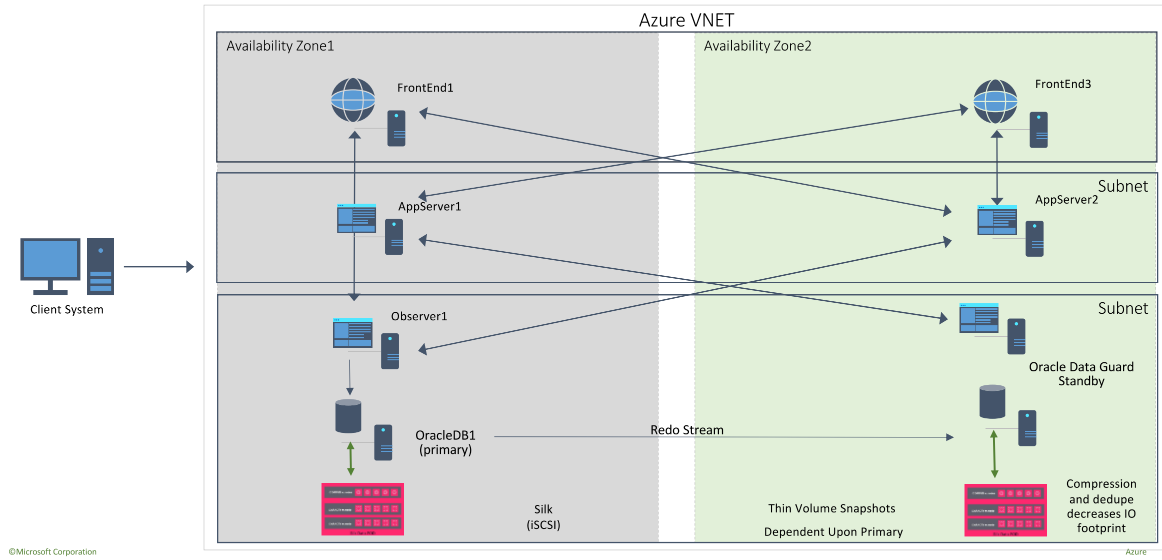
On a higher level, it's important to recognize that Oracle applications consist of multiple services, which can be hosted on either the same virtual machine or distributed across multiple virtual machines within the Azure environment. When migrating an Oracle workload to Azure, the transition of the application tier typically involves adapting the infrastructure to match the existing on-premises setup. This is usually followed by enlisting the expertise of Oracle on Azure specialists to facilitate the more complex migration of the database tier, often involving collaboration with adjacent teams.

While the demands on Oracle's application and middleware tiers are notably less pronounced, special attention must be given to the database per-core licensing model, which places significant emphasis on right-sizing the resources during the migration to the cloud. This consideration applies even to application tiers running on substantial, engineered Exalogic systems with a virtualized Oracle Virtual Manager (OVM) layer. In such cases, transitioning to Azure's virtual machines is a straightforward process, but the database layer presents more formidable challenges.

Creating Oracle Application instances within Azure follows best practices for Infrastructure as a Service (IaaS) workloads. This includes the option to utilize private or public endpoints for connectivity post-migration to Azure. Both Microsoft and Oracle recommend the implementation of a bastion host virtual machine with a public IP address, set up in a dedicated subnet to manage the application environment.

To enhance security, the establishment of network security groups at the subnet level is advised. This ensures that only specified ports and IP addresses are allowed for communication. For instance, the middle tier machines should exclusively receive traffic from within the virtual network, preventing direct external access to these machines.

Reference architecture – Dual Availability Zone Architecture with Silk



For high availability, you can set up redundant instances of the different servers in the same availability set or different availability zones. Availability zones allow you to achieve a 99.99% uptime SLA, while availability sets allow you to achieve a 99.95% uptime for the database tier in-region.

Choices

Once [subscription, tenant, etc. setup] for your organization, then it's important to make decisions around Operating Systems to support relational and applications in Azure.

When considering Operating Systems like Oracle Linux within the context of Azure, there are no licensing expenses involved. The costs primarily revolve around obtaining support to maintain its usage within the Azure environment. Therefore, if customers are not exclusively committed to Azure Monitor for overseeing their Linux Virtual Machines (VMs), it is advisable to contemplate Oracle Linux or RedHat. It's important to note that while Oracle Linux carries no additional licensing charges, RedHat does involve extra licensing costs.

Although Windows is less popular to run Oracle on, there's no better place to support Oracle running on Windows than Microsoft Azure. There are also options for non-production workloads on SLES, Ubuntu and other open-source operating systems.

Bastion Services

The bastion host serves as an optional element that can be utilized as a jump server for reaching the application and database instances. The virtual machine (VM) hosting the bastion serves can be configured with a public IP address, though it is recommended to establish a more secure connection using an [ExpressRoute] setup or a site-to-site VPN with your on-premises network. Moreover, it's advised to exclusively open SSH (port 22, Linux) or RDP (port 3389, Windows Server) for incoming traffic. For enhanced availability, you can deploy a bastion host across two availability zones or within a single availability set.

You also have the option to enable SSH agent forwarding on your VMs, enabling you to access other VMs within the virtual network by relaying credentials from your bastion host. Alternatively, you can employ SSH tunneling to access other instances.

To see an example of SSH agent forwarding, here's the following:

ssh -A -t user@BASTION_SERVER_IP ssh -A root@TARGET_SERVER_IP`

This instruction initiates a connection to the bastion server followed by an immediate SSH command, granting you access to a terminal on the intended target instance. In cases where your cluster setup varies, it might be necessary to define a user other than 'root' on the target instance. By utilizing the -A parameter, the agent connection is tunneled, allowing your local machine's private key to be employed seamlessly. It's important to note that agent forwarding functions as a sequence, thus the subsequent SSH command also integrates the -A flag. This ensures that any successive SSH connections initiated from the target instance also leverage your local private key.

Migrations are More Than a Database

The application tier is isolated in its own subnet. There are multiple virtual machines set up for fault tolerance and easy patch management. These VMs can be backed by shared storage, which is offered by Azure NetApp Files (ANF) and/or Premium SSDs. This configuration allows for easier deployment of patches without downtime. The machines in the application tier should be fronted by a public load balancer so that requests to the EBS application tier are processed even if one machine in the tier is offline due to a fault.

Load Balancer

An Azure load balancer allows you to distribute traffic across multiple instances of your workload to ensure high availability. In this case, a public load balancer is set up, because users are allowed to access the EBS application over the web. The load balancer distributes the load to both machines in the middle tier. For added security, allow traffic only from users accessing the system from your corporate network using a site-to-site VPN or ExpressRoute and network security groups.

There are a significant number of configurations and HA options for an Azure load balancer that can support various application configurations and requirements. If a load balancer doesn't meet the needs of the application, an application gateway or an Azure Route Server.

Oracle Built Right

This tier hosts the Oracle database and is separated into its own subnet. It is recommended to add network security groups that only permit traffic from the application tier to the database tier on the Oracle-specific database port 1521.

Microsoft and Oracle recommend a high availability setup. High availability in Azure can be achieved by setting up two Oracle databases in two availability zones with Oracle Data Guard. Clearly understand the difference between how we architect for a cloud environment and the choices made for an on-premises datacenter solution. Where RAC may justify in an on-premises data center, it tends to be much less valuable in a 3rd party cloud. Even if it was useful, Oracle will not support RAC in any public cloud. On top of this, the Azure cloud High Availability (HA) architecture solutions are often in contradiction with what RAC offers, creating a nonsensical solution.

RAC is often a marketing opportunity for Oracle. RAC must be acknowledged as an instance resiliency and scalability product, often not meeting many basic HA requirements. It is A solution, not THE solution and rarely do find workloads that require it for scaling, as well as benefit on savings on resources and price for customers once cloud architectural differences are realized.

1. Oracle only supports RAC in Oracle Cloud or on-premises and will refuse support in any third-party cloud environment, including Bare Metal.
2. Choose Oracle Data Guard for DR and HA, as it is very complementary to Azure HA design, just as Always-on AG is for SQL Server. We deploy the DG Broker, the observer and configure Fast-Start Failover to automate any failovers and manual switchovers and the DBMS_Rolling package will allow for online patching and upgrading.

How to Right-Size Oracle

Disclaimer: *Each version and database type of the Automatic Workload Repository (AWR) report can display data differently. The fields are the same, but the data may be in a different order, have a different header, etc. This document is to offer guidance in filling it out. If unsure, escalate for assistance, as an incorrect number could impact sizing estimates if not performed correctly.*

Assumptions

- AWR Report with 1-day or longer workload report
    - Ideally the report should cover peak load times
- The AWR Analysis sizing template
- Basic understanding of AWR data and Excel
- The Oracle database is either a single Oracle instance or RAC
- The Oracle database isn’t on an engineered system such as Exadata

Links to Worksheet

There is a traditional worksheet and instructions that many partners use that is connected to Microsoft Azure’s Github and can be downloaded by anyone interested in right-sizing Oracle workloads and translate them to Azure cloud infrastructure sizing.

Oracle AWR to Azure IaaS Worksheet: Sizing Worksheet Template

Detailed Instructions: GitHub Instructions

Updates are made regularly to the worksheets and instructions to reflect changes as necessary as the cloud evolves.

Sizing Process

Although the AWR report can provide essential data about workload, database usage and optimization for a cloud project, specific calculations can offer us invaluable data on what is required for an Azure IaaS VM to run the Oracle database in the cloud. The following will explain step by step what values to gather from the report and where to place them in the spreadsheet.

The Spreadsheet is broken down into two worksheets, the AWR and the Calculations worksheet. There are multiple lines to take RAC and multiple instances into consideration.

How to Read an AWR Report

Take the time to collect the correct information and fill out the right information in the AWR section of the template. Most miscalculations are due to incorrect data filled into the sizing assessment.

Most information that is required to sized out the workload will be found in the first page of the report. To begin, let's start with the first three columns:

1. **DBName:** This is the Global Name of the database:

Id	Name	Unique Name
160158887	DBPROD	DBPROD1

2. **Instance Name:** This is the unique name for the database instance or instance in Real Application Cluster(RAC) in Oracle.
3. **Host Name:** This is the name of the host and should be unique for each RAC instance.

The combination of DBName, Instance Name and Host name should be unique for the worksheet to calculate each database individually and then aggregate to a single database instance in the final calcuations.

Database Instances Includec

- Listed in order of instance nu

I#	Instance	Host
1	INST01	dbinst15.domain.com
2	INST02	dbinst16.domain.com

Elapsed Time and DB Time: The time the database consumed in minutes during the duration of the report and the time that elapsed during the report snapshot(s).

Report Total (minutes)	
DB time	Elapsed time
14,628.65	10,040.28

DB CPUs: DBCPUs can be found in numerous sections in the report, but the value that is needed is the one calculating DBCPUs by seconds(s) during the elapsed time. The number will most likely be the largest number reported in the assessment.

Time Model

I#	DB Time (s)	DB CPU (s)
1	2,048,549.15	1,463,292.19
2	2,092,224.28	1,476,583.93
Sum	4,140,773.43	2,939,876.11
Avg	2,070,386.71	1,469,938.06
Std	30,882.98	9,398.68

CPUs/Cores: Both the CPUs and core count for each instance are listed in the AWR report. This grants the a limited view on how many processor licenses should be licensed for the on-premises environment and a target for the cloud licensing to keep in the same cost. If the workload is resource constrained or there is special licensing conditions, there may be a requirement to increase the number of processor licenses to run the workload in the cloud, but this is less common unless the cloud solution is over-architected.

Always check to verify the VM SKU chosen is using a hyperthreaded vCPU to ensure the calculation for core licensing the customer will bring over to Azure is correctly calculated. If hyperthreading isn't on or has been turned off by Microsoft Support, then licensing from on-premises to Azure is a simple 1:1 cost per Document 2688277.1 (oracle.com):

DETAILS

For the purposes of this document, Non-Oracle Public Cloud Environments are defined as:

- (a) Non-Oracle Public Clouds. Examples: Google Cloud Platform, Amazon AWS, Microsoft Azure, IBM Cloud, Alibaba Cloud, etc.
  or
- (b) Environments that are in any way considered an extension of Non-Oracle Public Clouds including but not limited to running Non-Oracle cloud management software, cloud billing, cloud support, cloud automation, cloud images, or cloud monitoring. Examples: Google Bare Metal Solution, Amazon AWS Outpost, Microsoft Azure Stack, IBM Bluemix Local, Alibaba Hybrid Cloud, etc.

Support Policy for Non-Oracle Public Cloud Environments

Oracle has not certified any of its products on Non-Oracle Public Cloud Environments. Oracle Support will assist customers running Oracle products on Non-Oracle Public Cloud Environments in the following manner: Oracle will only provide support for issues that either are known to occur on an Oracle Certified Platform outside of a non-Oracle Cloud Environment (Oracle Certification Home), or can be demonstrated not to be as a result of running on a Non-Oracle Public Cloud Environment.

If a problem is a known Oracle issue, Oracle support will recommend the appropriate solution on an Oracle Certified Platform outside of a non-Oracle Cloud Environment. If that solution does not work in the Non-Oracle Public Cloud Environment, the customer will be referred to the Non-Oracle Public Cloud vendor for support. When the customer can demonstrate that the Oracle solution does not work when running on an Oracle Certified Platform outside of a non-Oracle Cloud Environment, Oracle will resume support, including logging a bug with Oracle Development for investigation if required.

If the problem is determined not to be a known Oracle issue, we will refer the customer to the Non-Oracle Public Cloud vendor for support. When the customer can demonstrate that the issue occurs when running on an Oracle Certified Platform outside of a non-Oracle Cloud Environment, Oracle will resume support, including logging a bug with Oracle Development for investigation if required.

Support Policy for Oracle Real Application Clusters (RAC)

Oracle does not support Oracle RAC or Oracle RAC One Node running on Non-Oracle Public Cloud Environments.

I#	Num CPUs	CPU Cores
1	320	160
2	320	160
3	320	160

In the example above, the 3-node RAC has 320 CPUs, at 160 Cores, each licensed at .5 processor license each. With this knowledge, we can estimate there are 480 processor licenses that can be used in the Azure Cloud for vertical scaling on a single instance and for the secondary Oracle DataGuard standby.

Memory (GB): Memory can be captured multiple ways in the AWR, either on it's own or part of the SGA/PGA information-

Memory (M)
6,191,158.41

/ 1024= Correct Value for Spreadsheet

%Busy CPU: This value is clearly stated in the report and is used to identify CPU saturation. A CPU is either on or off, but to know if enough CPU is available is part of our estimates. For Oracle container databases, this value isn't listed and it can cause some confusion. The value is listed in other AWR report in the OS Statistics and for each instance CPU totals, look for %Busy.

% Busy
25.84

SGA(MB): This can be under different tables, depending on the version. It can be a good idea to do a search for "SGA". SGA Target demonstrates the beginning and end values for an adjusting value. If you use this section, take the highest of the two values, (peak). If no value is shown for an ending value, it means no adjustment was made from the beginning value.

Sga Target	
Begin	End
32,768	

PGA(MB): Is the Process Global Area and this is a specialized area of memory allocated for sorting, hashing and other important processing. Heavier sorting is performed in Oracle due to lacking clustered indexes in the Oracle design. The memory allocated may not meet the needs of the database, which is a resiliency vs. sizing issue. Like SGA, the PGA Target will display a beginning and ending value for some AWR Reports. Take the larger of the two values displayed.

PGA Target	
Begin	End
32,768	

Read Throughput (MB/s) and Write Throughput (MB/s): This is a value that can be displayed in multiple ways and sections in the AWR report depending on the version and type of Oracle product. Search the report, (find on page if in a browser) for "IO Statistics". For the example below, a RAC database with 3 nodes displays the Read throughput and write throughput for each instance:

IOStat by Function

		Requests				MB					
I#	Function Name	Total	IOs/s	Reads/s	Writes/s	Total	IO MB/s	Read MB/s	Write MB/s	Waits: Count	Avg Time
*	Total	1,581,621,029	29,293.53	19,839.48	9,454.06	452,920,597	8,388.64	6,568.84	1,819.80	519M	507.38us
	RMAN	117,080,157	2,168.47	1,895.60	272.86	126,061,919	2,334.82	1,790.06	544.76	294K	.99ms
	Smart Scan	120,087,660	2,224.17	2,224.17	0.00	100,626,420	1,863.72	1,863.72	0.00	0	
	Others	142,617,748	2,641.45	2,347.72	293.73	70,587,460	1,307.37	1,035.42	271.95	81M	772.34us
	Streams AQ	65,466,095	1,212.51	1,212.51	0.00	68,204,442	1,263.23	1,263.23	0.00	876K	6.88ms
	LGWR	79,729,417	1,476.69	0.13	1,476.55	27,737,403	513.73	0.00	513.73	12.4K	818.45us
	Direct Reads	33,626,408	622.80	553.19	69.61	17,107,441	316.85	282.22	34.63	0	
	Buffer Cache Reads	624,211,846	11,561.16	11,561.16	0.00	16,889,994	312.82	312.82	0.00	436.7M	445.09us
	Direct Writes	56,913,309	1,054.10	44.95	1,009.15	14,721,412	272.66	21.36	251.29	8979	640.09us
	DBWR	341,886,723	6,332.16	0.01	6,332.14	10,984,057	203.44	0.00	203.44	645	573.64us
	Data Pump	1,666	0.03	0.03	0.00	49	0.00	0.00	0.00	934	676.66us
1	Total	726,133,793	13,448.87	9,231.35	4,217.53	248,993,562	4,611.66	3,738.47	873.19	204.9M	522.33us
	RMAN	70,066,122	1,297.71	1,144.31	153.40	74,193,226	1,374.15	1,067.96	306.19	177.3K	.99ms
	Smart Scan	68,800,451	1,274.27	1,274.27	0.00	60,816,059	1,126.39	1,126.39	0.00	0	
	Streams AQ	39,765,909	736.51	736.51	0.00	41,358,808	766.02	766.02	0.00	523.2K	6.07ms
	Others	62,115,807	1,150.46	1,027.04	123.42	32,428,062	600.61	486.51	114.10	33.8M	735.56us
	LGWR	32,315,706	598.53	0.07	598.45	12,034,230	222.89	0.00	222.89	6855	829.32us
	Direct Reads	17,122,289	317.13	284.93	32.20	8,617,683	159.61	142.46	17.15	0	
	Buffer Cache Reads	257,165,089	4,763.01	4,763.01	0.00	7,998,854	148.15	148.15	0.00	170.4M	462.48us
	Direct Writes	22,510,719	416.93	1.17	415.75	6,333,887	117.31	0.99	116.32	180	2.00ms
	DBWR	156,270,035	2,894.31	0.00	2,894.31	5,212,704	96.55	0.00	96.55	258	810.08us
	Data Pump	1,666	0.03	0.03	0.00	49	0.00	0.00	0.00	934	676.66us
2	Total	855,487,236	15,844.66	10,608.13	5,236.53	203,927,035	3,776.98	2,830.36	946.61	314.1M	497.63us
	RMAN	47,014,035	870.76	751.29	119.46	51,868,693	960.67	722.10	238.57	116.7K	.98ms
	Smart Scan	51,287,209	949.90	949.90	0.00	39,810,361	737.34	737.34	0.00	0	
	Others	80,501,941	1,490.99	1,320.68	170.31	38,159,398	706.76	548.91	157.85	47.2M	798.65us
	Streams AQ	25,700,186	476.00	476.00	0.00	26,845,634	497.21	497.21	0.00	352.8K	8.08ms
	LGWR	47,413,711	878.16	0.06	878.10	15,703,173	290.84	0.00	290.84	5522	804.96us
	Buffer Cache Reads	367,046,757	6,798.15	6,798.15	0.00	8,891,140	164.67	164.67	0.00	266.4M	433.97us
	Direct Reads	16,504,119	305.68	268.26	37.41	8,489,758	157.24	139.76	17.48	0	
	Direct Writes	34,402,590	637.18	43.78	593.40	8,387,525	155.35	20.37	134.98	8799	612.34us
	DBWR	185,616,688	3,437.85	0.01	3,437.84	5,771,353	106.89	0.00	106.89	387	416.02us

Caution- *In Oracle 12.1 and 19c versions, there were some issues around IOPs(calculations for requests) in the functions summary section. As we focus so heavily on MBPs(throughput) the Oracle SMEs didn't worry too much about it, but it could be alarming if you pay attention to these values. Simply go to the next section from the function summary to the summary by file type and the values for requests will be corrected.*

Calculating Factors for Worksheets

Once you've filled in this information, note that there is a gray box below the area to enter in all sections, for instance:

Name of "fudge factor" adjustable	Default value	Setting
Est'd Peak CPU factor	2.00	2.00
Est'd Peak RAM factor	2.00	1.10
Est'd Peak I/O factor	2.00	3.00
vCPU HT multiplier	2	2
%Busy CPU-thrashing threshold	0.75	0.75
%Busy CPU-thrashing multiplier	1.25	1.25

Please do not change the default values

These values are here to help calculate the type of workload that you are bringing over. For most workloads, the defaults should be retained, but if you understand the workload, some adjustments can be made to build out the most effective peak workload from the AWR collection.

For Exadata, an IO metric fudge factor should be higher, and in most Exadata workloads the IO factor number is increased from 2.00 to 4.00-6.00 to take the higher IO into consideration from loss in offloading and other engineered features.

Decide what you want for each of the following and make changes based on the following:

Peak CPU Factor: 2.00 is standard, 4.00 is for a workload that might have a huge variance expectation once it goes to the cloud, such as AIX or Sparc.

Est'd RAM Factor: Same for CPU, but for RAM estimate. Normal is 2.00, 4.00 would be normal for an Exadata where the SGA is commonly shrunk to promote offloading. Once decoupled from Exadata, running more in memory will allow for better performance and not rely on IO as often.

vCPU HT Factor: Commonly 2.00 and this should be the default going to IaaS Azure VMs. If the AWR snapshot is for a peak workload and doesn't require a peak simulated into the assessment, this number could be decreased, but should never be decreased below 1.0.

Busy CPU Thrashing Threshold: This is set at 75, which is the percentage that over, signals the on-premises host is saturated and needs to be sized up to handle the workload. This type of workload should expect a need to increase Oracle processor licenses, as well.

IO metrics (IOPS & MB/s) fudge factor: 2.00 is for transactional system, 4.00 is for DSS/OLAP, 6.00 is for Exadata.

Calculations Spreadsheet

Don't fill in any area OUTSIDE of the fields instructed, which have headers **filled with blue.** Columns are dependent on what is filled in on the AWR page to match what is in the appropriate fields on the Calculations page.

1. Enter the DB Name and Instance Name, duplicating the DB name, if necessary, that corresponds to the instance name. Do not leave the first column blank if you fill in the second and the first value for each RAC database should be the global database name, where the instance name should be unique for that instance.

2. Although the column looks like it extends for two, place the hostname for the servers for every instance in the first column of the next section. This section will help to identify how workloads are distributed across hosts and assist in consolidation efforts from the on-premises systems to the cloud.
3. Enter the global database name for each assessed Oracle database in the next section. Remember, all RAC databases will be aggregated to a single instance and sized for what it will require to run them in Azure as a single instance. In other words, each database will be listed only ONCE here.

While inputting the values into the second worksheet, you'll observe calculations coming up. Once this step is finished, you will possess data for sizing the workload into Azure for each database. Subsequently, these figures will furnish you with the necessary insights to select one or more IaaS Azure VMs for devising an appropriately sized solution for the Oracle customer.

Example of an Oracle Sizing Assessment

Here is an illustration of the outcome derived from a customer interaction. This pertains to a scenario where two databases are in use, each functioning within a 2-node RAC environment. It's worth noting that the DB Name column is duplicated for both databases, while the instance name remains distinct. The rest of the details were left unfilled, as the preceding worksheet's content is automatically imported and processed to fulfill the required calculations.

DB Name	Instance Name	%DB Time of Elapsed Time	%DB CPU of server capacity	Total ORA (GB)	Total IOPS	Total Throughput (MB/s)	Est'd Azure vCPUs
DBPROD1	dbprod1	3355.655%	43.229%	31	3,775.15	1,318.36	25.17
DBPROD2	dbprod2	820.037%	27.292%	131	842.90	23.10	6.15
DBPROD3	dbprod3	2784.107%	30.759%	17	4,407.71	70.90	20.88
DBRAC	dbrac11	381.437%	4.836%	15	290.31	776.55	2.86
DBRAC	dbrac12	507.727%	6.240%	10	303.72	475.46	3.81
DBPROD4	Cell Styles	892.448%	10.220%	7	1,336.10	3,622.24	6.69
		8741.412%	122.575%	210	10,955.89	6,286.61	65.56

In the second segment, solely the name of the host was filled into the initial column for every node representing the RAC instances. With two nodes assigned to each of the two databases, a total of four entries were included, and these values were sourced from the initial worksheet.

Aggregated calculations by host

Host	Name	%DB Time of Elapsed Time	%DB CPU of server capacity	Total ORA (GB)	Total IOPS	Total Throughput (MB	Est'd Azure vCPUs
host1		3355.655%	43.229%	30.84	3775.15	1318.36	25.17
host2		820.037%	27.292%	130.76	842.90	23.10	6.15
host3		2784.107%	30.759%	17.03	4407.71	70.90	20.88
rachost1		381.437%	4.836%	14.84	290.31	776.55	2.86
rachost2		507.727%	6.240%	10.45	303.72	475.46	3.81
host4		892.448%	10.220%	6.52	1336.10	3622.24	6.69
Total		8741.412%		210.44	10955.89	6286.61	65.56

In the third and last section, each of the database names were filled in and with this information, the calculations for the current workload, along with simulated peak for each- memory, IO and vCPU is calculated for sizing.

Aggregated calculations by database

DB Name	%DB Time of Elapsed Time	Total vRAM (GiB) consumed only by Oracle	Est'd Azure vRAM for server	Total IOPS	Total Throughput (MB/s)	Est'd Azure IOPS for peak load	Est'd Azure Throughput (MB/s) for peak load	Est'd Azure vCPUs for avg load	Est'd Azure vCPUs for peak load	DB Size in TB	Bkup Info in TB
DBPROD1	3355.655%	31	46	3,775.15	1,318.36	7,550.30	2,636.72	26	39	80.00	320.00
DBPROD2	820.037%	131	196	842.90	23.10	1,685.80	46.20	7	11	7.00	28.00
DBPROD3	2784.107%	17	26	4,407.71	70.90	8,815.42	141.80	21	32	0.80	12.00
DBRAC	444.772%	25	38	594.03	1,252.01	1,188.06	2,504.02	7	11	2.50	18.00
DBPROD4	Cell Styles 448%	7	10	1,336.10	3,622.24	2,672.20	7,244.48	7	11	4.00	30.00
Total	8297.02%	210	316	10,955.89	6,286.61	21,911.78	12,573.22	68	102	94.30	408.00

For example:

DBPROD1 will require:

26 vCPU for an average load and 39 vCPU for a max workload.

A server with 46G of memory and 31G allocated to the database.

Disk IOPS 7550 and 2637MB/s throughput.

There is a total that is displayed at the bottom, but this is only available if you need to know how many resources will be required for the project. The value we have here is what we require to size out the Azure VM.

Calculations can be seen for the rest of the databases and using this information, it is easy to size out exactly what target hardware to accomplish the needs of the workload.

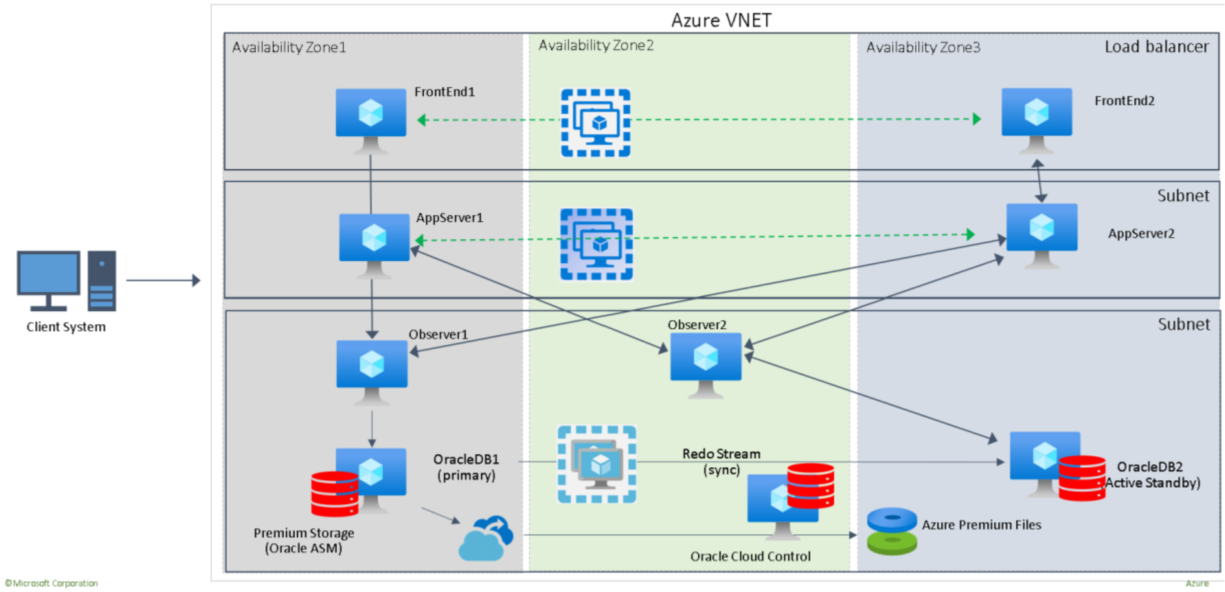
Using this information, VMs, backup solutions, OS disk and datafile storage can be assessed. As the example is that of a large Oracle environment, (93TB) storage is one of the largest cloud expenditures to meet the needs of the Oracle workloads.

Possible recommendations:							
Calculated vCPU x vRAM:				**Monthly pricing**			
Suggested Instance types:		**Azure Instance type**	**vCPU x vRAM**	**Pay-as-you-go**	**1-year**	**3-year**	**Storage Req.**
DBPROD1		**E32ds v5**	*32vCPU * 256G*	2,628.00	2,190.89	1,793.59	Silk
DBPROD2/DBPRO4		**E20ds v5**	*20vCPU * 160G*	1,854.20	1,369.35	1,120.99	Silk
DBPROD3		**E32ds v5**	*32vCPU * 256G*	2,628.00	2,190.89	1,793.59	P30
DBRAC		**E16bds v5**	*16vCPU * 128G*	1,641.04	1,188.53	956.69	Silk
Azure Backup		**4 Hosts Volume Snapshots**		7,760.60	7,760.60	7,760.60	
OS Disk		**P10 * 4**	*128G each*	78.84	78.84	78.84	
Premium SSD		**P30 * 1**	*1TB each*	135.17	135.17	135.17	
Silk		**93TB**	*dNodes/4 cNodes*	37,401.42	37,401.42	37,401.42	

Choosing the Correct VMs and Storage

High Level Oracle on Azure for IaaS

Oracle High Availability in the Azure cloud marries Azure High Availability with Oracle Data Guard to create solutions that use many of Oracle's Maximum Availability Architecture advanced concepts. Due to the differences between on-premises architecture and the public cloud, there are significant differences. Where Data Guard is more focused on Disaster Recovery in an on-premises solution, in Azure, it's front and center for High Availability, leveraging Fast-Start Failover, the DG Broker, Observer, etc.



Decisions around Cross-region deployments, Availability Zones, or Availability Sets, along with number of Data Guard standbys in a specific customer environment is based on Service Level, Recovery Point Objective, (RPO) and RTO, (Recovery Time Objective). This information will also provide the information required for backup and recovery strategies and storage requirements, (storage often has features that provide value in these focus areas.)

The above, classic highly available and 99.996% uptime for Oracle is a recommendation to begin with and from here, the architecture can simplify or evolve.

Azure recommendations for Oracle Virtual Machines

Below is some typical Oracle VM configuration checklist items

Type	Source	Azure Recommendation
Storage	https://docs.microsoft.com/en-us/azure/virtualmachines/windows/premium-storage#scalabilityand-performance-targets	Start with Premium SSD or PV2 unless workload MBPs, (throughput) requires more or simplified management is desired.
	https://docs.microsoft.com/en-us/azure/virtualmachines/workloads/oracle/oracle-design#diskcache-settings	Separate redo logs from datafiles TS on separate data disk whenever log latency is experienced.

	https://docs.microsoft.com/en-us/azure/virtualmachines/linux/optimization#io-schedulingalgorithm-for-premium-storage	Set NOOP or Deadline algorithm for I/O scheduling
	https://docs.microsoft.com/en-us/azure/virtualmachines/windows/premium-storage#premiumstorage-for-linux-vms	Disable "barriers" for disks with cache read-only for datafiles and redo/FRA to none.
	https://docs.microsoft.com/en-us/azure/virtualmachines/windows/premium-storageperformance#disk-striping	Use Stripe size 64KB

Type	Source	Azure Recommendation
	https://docs.microsoft.com/en-us/azure/virtualmachines/workloads/oracle/oracle-design	For OS disks, use default Read/Write host level caching and use premium SSD, (P6-P15 recommended for Oracle VM)
Temp/Swapfile	[Ephemeral OS disks - Azure Virtual Machines | Microsoft Docs]()	Ensure swapfile for Linux or Windows is located on attached, ephemeral storage. Use a VM SKU that includes an attached temp storage.
	https://docs.microsoft.com/en-us/azure/virtualmachines/workloads/oracle/oracle-design	For DATAFILES, use Read-Only host level caching for Premium SSD available on P30, P40 or P50. For the P50, don't allocate the last 1G to stay under the max size of 4095G for host level caching.

Redo logs	https://docs.microsoft.com/en-us/azure/virtualmachines/workloads/oracle/oracle design#configuration-options	If latency is experienced, consider moving redo logs to a separate volume of storage.
	Ultra disks for VMs - Azure managed disks - Azure Virtual Machines | Microsoft Docs	Consider Ultra Disk for redo latency. Scaling feature as redo demands increase. High redo log parallel write may benefit from separating redo log members to individual ultra disks, (A members on one ultra disk, B members on second ultra disk.)
	https://docs.microsoft.com/en-us/azure/virtualmachines/linux/how-to-enable-write-accelerator	Enable Write Accelerator for Redo logs disks when using premium disk
	https://docs.microsoft.com/en-us/azure/virtualmachines/linux/how-to-enable-write-accelerator	Set Cache policy: None + Write Accelerator for Redo logs disks on premium SSD.
	https://docs.microsoft.com/en-us/azure/virtualmachines/linux/how-to-enable-write-accelerator	Use I/O sizes (<=32 KiB) (Redo block size < 32)
Network	https://docs.microsoft.com/en-us/azure/virtualnetwork/create-vm-accelerated-networking-cli	Use Accelerated Networking, dependent upon VM SKU choice for availability.
Oracle DB		if filesystem is ext4 use DB param:
		filesystemio_options=ASYNCH

		If using ASM, partitions used for ASM disks should be created with a 1MB (2048 sectors) offset
		If using ASM, set diskgroup au_size >= 4M for large databases
	https://docs.oracle.com/database/121/UNXAR/api_vlm.htm#UNXAR391	If using ASMM, enable HugePages on Linux and disable Transparent HugePages: https://oraclebase.com/articles/linux/c onfiguring-huge-pagesfor-oracle-on-linux-64 + HugePages on Oracle Linux 64-bit (Doc ID 361468.1)
Multi-tenant		Oracle Multi-tenant for Oracle relies on a container database with pluggable databases. Customers are allowed THREE pluggable databases with no additional charge, but more than FOUR PDBs often experience degradation. An additional container database, (CDB) should be allocated on the VM per each four pluggable databases, (PDBs).
I/O Scheduler	I/O scheduler to “deadline” or "none"	Limit impact to IO by the scheduler
File System optimization	Set barrier=0	This disables the use of write barriers in the jbd code by setting a flag in Linux.

Proximity Placement Groups	Address SQL*Net waits in AWR and network latency between resources in Azure	Oracle is a multi-tier system- all applications and sibling databases should be placed in a PPG so Azure is aware and allocates resources in close proximity to each other, eliminating network waits.

IO Is the Challenge in the Cloud

Unlike on-premises, high vCPU count and memory is available in the Azure Cloud. Rarely do Oracle workloads run into a challenge on achieving the vCPUs required after a sizing assessment. As Oracle charges licensing by core, this is another reason to perform a right-sizing assessment before recommending a VM SKU for the Oracle database workload.

High Memory Shouldn't be the Default for Oracle

Azure VM SKUs have four main categories for relational workloads:

- General Purpose
- Compute Optimized
- Memory Optimized
- Storage Optimized

Of these, only **Memory Optimized** are primarily used for Oracle workloads. The Memory Optimized VM SKU category includes SKUs from the D, E and M series virtual machines. For Oracle, as discussed earlier in the recommended checklist, there are two areas that we drill down on for preferred SKU series:

- E-series, Eds v4 or v5 is highly recommended
    - Allows for Premium SSD for OS Disk.
    - Has ephemeral storage to be used for swap.
    - V5 has great performance enhancements and available everywhere.
    - High IO limits, upwards of 12GBPs for Silk on some SKUs.
    - Also consider the Ebds v5 series VM for higher IO limits for less vCPU.
- M-series
    - Great for high memory requirements
    - Enhanced vCPU performance
    - Lower IO limits, host level caching is low for attached storage vs. E-series.
    - Has ephemeral storage to be used for swap.
    - Offers accelerated networking options.
    - More limited for attached storage solutions.
    - Watch for the V3 coming out in late in 2023.

The Oracle database performances are strictly influenced by the following parameters:

- Disk throughput, (MBPs)
- Read/write IOPs
- Network latency
- CPU, RAM

IO Limitations in the Public Cloud

In the cloud, its essential that physical resources aren't overallocated once they become virtualized. One of the most common ways to do this is to impose limits at the service level which results in simplified management and assurance that each customer will have the resources they have been promised by service level agreements (SLAs).

When discussing storage input/output (IO), it's important to consider requests, also known as IOPs. However, the significance of IOPs as a measure of IO capability depends on the consistency and known size of these requests. In the context of Oracle, the value of IOPs might be diminished, and it becomes crucial to grasp throughput, often referred to as megabytes per second (MBps).

In the case of Azure, the commonly used storage solution, Standard SSD, is generally inadequate for demanding IO workloads such as Oracle. In contrast, premium SSD serves as the starting point for intricate relational systems. For scenarios involving Oracle on Azure outside of SAP, the selection of storage should be aligned with the IO requirements of the workload. This choice should then be coupled with an appropriate Virtual Machine (VM) SKU with an IO limit set sufficiently high to accommodate the peak demands of the Oracle database.

To successfully migrate Oracle, it's imperative to comprehend the limitations of first-party storage solutions within Azure. This understanding forms a fundamental initial step in the process of achieving a successful Oracle migration.

Workload	Premium SSD	Ultra Disk	ElasticSAN	Silk PV2	Silk Traditional
Availability	All Regions	All Regions	Limited	X	Where L-Series Available
IO up to 750MBps	X	X	X	X	X
IO to 2GBps	If Striped	X	X	X	X
IO to 10GBps				X	X
Isn't Throttled by Low Storage Limits at VM			X	X	X
Compression/Dedupe				X	X
Thin Snapshots	X	Private Preview		X	X
Thin Clones				X	X
Scaling Options		X	X	X	X
Cost Scale	$	$$-$$$$	$$$	$$	$$$

In the storage matrix above, initial cost can be low for premium SSD, but IO throttling at the VM level can limit when it is useful. For Ultra Disk, the limit is still imposed and the cost can quickly increase to higher than high IO, network attached solutions without strict limits.

Storage in Azure

When you initiate the creation of a new managed disk using the portal, you will be given the option to choose the Account type that corresponds to the desired variant of the disk. It's important to note that not all available disk types are visible in the drop-down menu. For Oracle workloads, the recommended selection for lower-performing storage is Premium SSD, but it is common as true workload demands arise, network attached storage solutions must be used. For greater scalability, options like Silk come in to win customer workloads, ensuring they have similar experience to what was achieved on-premises. To assist in making the best choice, a feature comparison matrix above offers a limited overview, helping align the Oracle workload with the appropriate disk type.

After configuring storage on a virtual machine (VM), it's a prudent practice to subject the disks to a load test before establishing a database. This helps in understanding the I/O rate, encompassing both latency and throughput, which is crucial for assessing whether the VMs meet the expected throughput while also adhering to latency targets.

Numerous tools are available for conducting application load testing, such as SLOB (Silly Little Oracle Benchmark), Oracle Orion, Swingbench, and FIO. However, it's worth noting that due to limited community support, the open-source product HammerDB is less recommended for Oracle IO testing. Additionally, when comparing platforms, it's important to mention that metrics like Transactions per Minute (TPM) aren't as easily accessible as they are for SQL Server or MySQL.

Once an Oracle database is deployed, it is advisable to rerun the load test. This involves simulating both regular and peak workloads to establish a performance baseline for your environment.

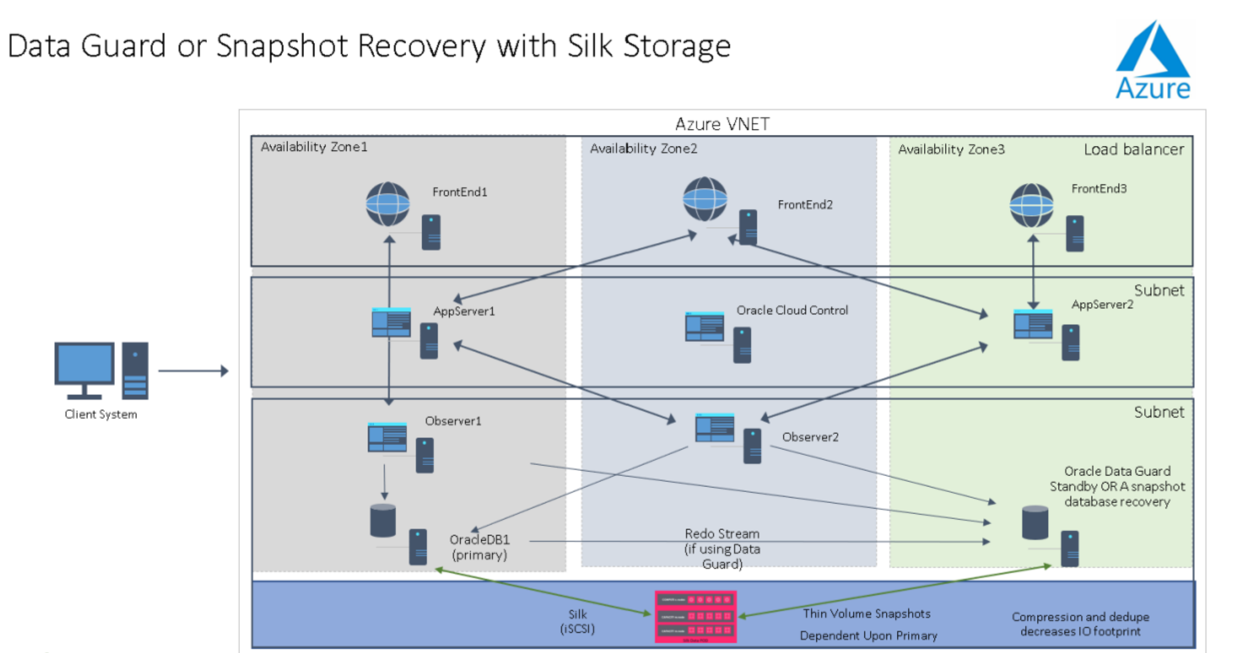
Consistently prioritizing throughput over Input/Output Operations per Second (IOPs) in relation to storage size is crucial, with an emphasis on Megabytes per Second (MBPs). For instance, if you require 750 MBPs but only need 200 GB of storage, it might still be beneficial to opt for the P40 class premium disk, even if it offers 1000 GB of storage. This approach ensures that the MBPs requirement is met, especially when host-level read-only caching is enabled.

The MBPs/IOPS ratio can be determined from the sizing assessment based on the AWR report. This ratio is calculated by considering factors like redo log activity, physical read and write rates obtained from the AWR report. In scenarios involving larger window workloads, aggregations and averages are factored in to provide a more accurate understanding of peak IO workloads.

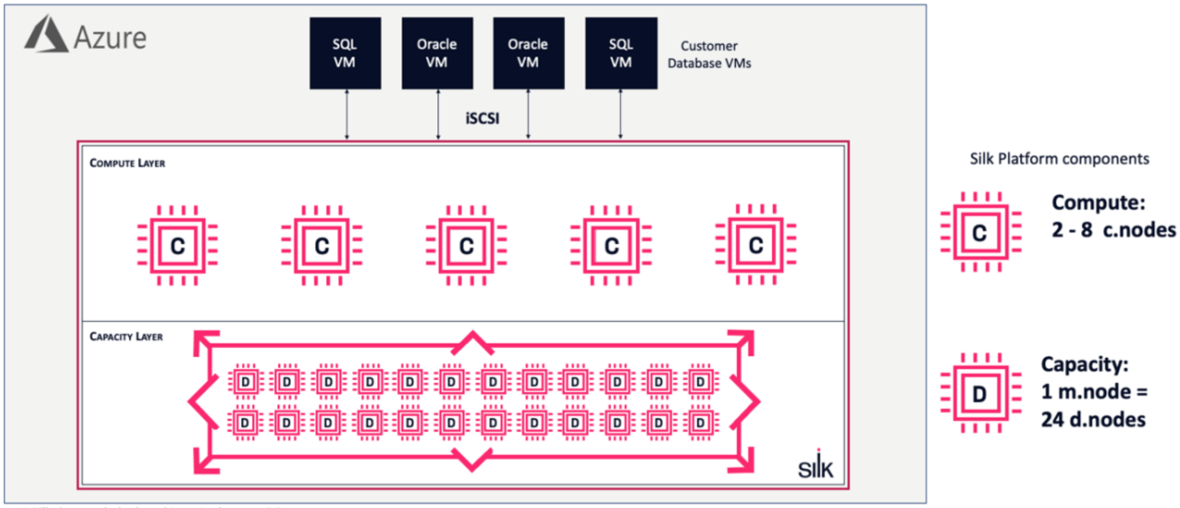
Silk

Keep in mind that Silk is all Azure for the hardware that is part of its solution. A Silk data pod uses Kubernetes at the core of its technology. The high IO solution mirrors the architecture of traditional on-premises SAN transparently to the compute layer, but internally is using D-series for controllers and L-series Azure Virtual Machines as the disk arrays or Premium SSD V2.

When using L-Series VMs configured as part of a Silk data pod and providing high IO performance using the ephemeral disk attached to the VMs, additional features assist in adding to the total cost of ownership. With the addition of compression/dedupe, along with thin volume snapshot backups and thin cloning capabilities, this solution provides exceptional throughput at a lower cost than other solutions such as ultra disk or Azure Netapp Files (ANF).



Silk Platform Architecture



A Silk data pod, deployed in a single zone, RG

In the image above, the Silk data pod is presented as a storage layer to the database VM and is completely transparent to the database. The Oracle database simply views it as storage to use, just as it would any other disk, but the database is offered significant performance advantages with the virtualized storage layer.

Metric	Azure + Silk	Azure High End	Gain vs. High End	Azure Native	Gain vs. Native
Read IOPS	1.5m+	160K	10x	20K	75x
Read BW	26 GB/s	4 GB/s	6.5x	0.9 GB/s	28x
Write IOPS	1.1M+	160K	6.3x	20K	55x
Write BW	13 GB/s	2 GB/s	6.5x	0.9 GB/s	14.4x

All performance numbers achieved at 1.5ms consistent latency or lower, with data services enabled

- Designed for mixed workloads – OLTP & BI
- Elastically scale performance up and down
- Patented algorithms for high parallelism
- Automatic tuning for optimal CX
- Shared performance for all applications

One of the Silk's greatest contributions is its value regarding total cost of ownership (TCO). Unlike simple storage, a Silk high IO solution not only meets the IO demands of Oracle workloads, but it also provides storage savings with its impressive data reduction capabilities. For multi-tier Oracle environments, like E-business suite, which is known for numerous monthly refreshes and clones, thin provisioning of the entire stack can provide impressive cloud cost savings and valuable time back for database administrators and developers.

Application consistent volume snapshots can relieve IO pressure on VMs to allow for additional sizing decrease, both on compute and Oracle licensing, which is often most Oracle migration costs and must license on the vCPU allocated on the VM.

Unified identity and access management

Microsoft has expanded the capabilities of Azure Active Directory (Azure AD) to encompass cloud functionality, facilitating seamless single sign-on for both enterprise and web applications situated in cloud environments. Leveraging cross-cloud connectivity, Oracle clientele can seamlessly incorporate access management utilizing the Azure AD infrastructure through a federated identity framework. This amalgamation establishes a cohesive mechanism for both authenticating and authorizing users and applications.

The advantages presented by federated identity are manifold, encompassing streamlined single sign-on processes, mitigated security vulnerabilities, and heightened organizational efficiency. Azure's reinforcement of security measures bolsters this endeavor. Enterprises can place their trust in a cloud ecosystem constructed with bespoke hardware, intrinsically integrated security protocols within hardware and firmware components, as well as augmented safeguards against perils such as Distributed Denial-of-Service (DDoS) attacks.

Key benefits include:

- Harnessing the cutting-edge, multi-layered security infrastructure upheld across Azure data centers on a global scale.

- Swiftly fortifying workloads through inbuilt controls and services within Azure, encompassing facets such as identity management, data protection, networking security, and application security.

- Timely identification of potential threats through distinctive intelligence mechanisms.

Benchmarking

The topic of comparing on-premises performance with cloud performance is almost inevitable in initial discussions. While a direct one-to-one comparison between dedicated on-premises hardware and cloud infrastructure isn't straightforward, there are benchmarking tools that offer some insights into the advantages of migrating relational workloads to the cloud.

While we frequently gather data concerning CPU and memory usage, the most informative metric for assessing successful cloud migrations is input/output (IO) performance. Although adjusting virtual CPUs and memory is relatively straightforward in cloud environments, managing storage might not be as uncomplicated, particularly when dealing with solutions like Oracle's Automatic Storage Management (ASM).

Jens Axboe developed the Flexible IO benchmarking tool, known as FIO, to facilitate adaptable testing of Linux I/O subsystems and schedulers. This singular testing capability that furnishes IO performance data applicable to various applications and workload simulations has proven highly beneficial to administrators, which accounts for its enduring popularity.

With a strong presence in the Linux industry, FIO is a collaborative effort involving over 5000 users. It remains accessible to anyone interested in benchmarking diverse I/O workloads.

- General documentation on FIO
- FIO on Github

- FIO Github
- FIO Workload Benchmark Examples

SLOB- i.e., Silly Little Oracle Benchmark, is often the go-to for Oracle specialists to perform Oracle specific benchmarks. It is an open-source tool, maintained by the Oracle community and comes with easy workload generation. If you'd like to know more about SLOB, check out the following links:

- General Info on SLOB
- SLOB GitHub
- SLOB Use Cases

Oracle Swingbench- is an Oracle specific benchmark tool developed by Dominic Giles, who has worked for both Oracle and Google. This tool is very Oracle specific, (as is SLOB) and well-known by Oracle specialists for measuring performance for Oracle workloads.

- General information on Oracle Swingbench
- Swingbench Installation

Recommended practices with IO Benchmark Tools

1. Anticipating identical performance within a virtualized environment is an impractical expectation. Define the desired response times and network or I/O latency for the workload objective. Remain adaptable in resource allocation and scalability to achieve optimal performance.

2. Utilize AWR/Statspack reports in conjunction with I/O benchmark outcomes. Frequently, challenges in performance stem from multiple factors contributing to latency. It is crucial to distinguish whether the root cause lies in Oracle optimizer, maintenance tasks, assumptions about workload, or if it is more often attributed to VM and storage selections or configurations.

3. Deconstruct performance concerns into more manageable lists. Refrain from upgrading both the database and application tiers simultaneously during the cloud migration process. Merging multiple projects can give rise to performance issues. Always opt to dissect and address each issue in sequence rather than tackling them as a unified challenge. In scenarios where multiple performance issues arise, fragment them into prioritized lists that can be systematically addressed, conquered, and ultimately resolved.

Migration Recommended Practices

Know Your Database Size

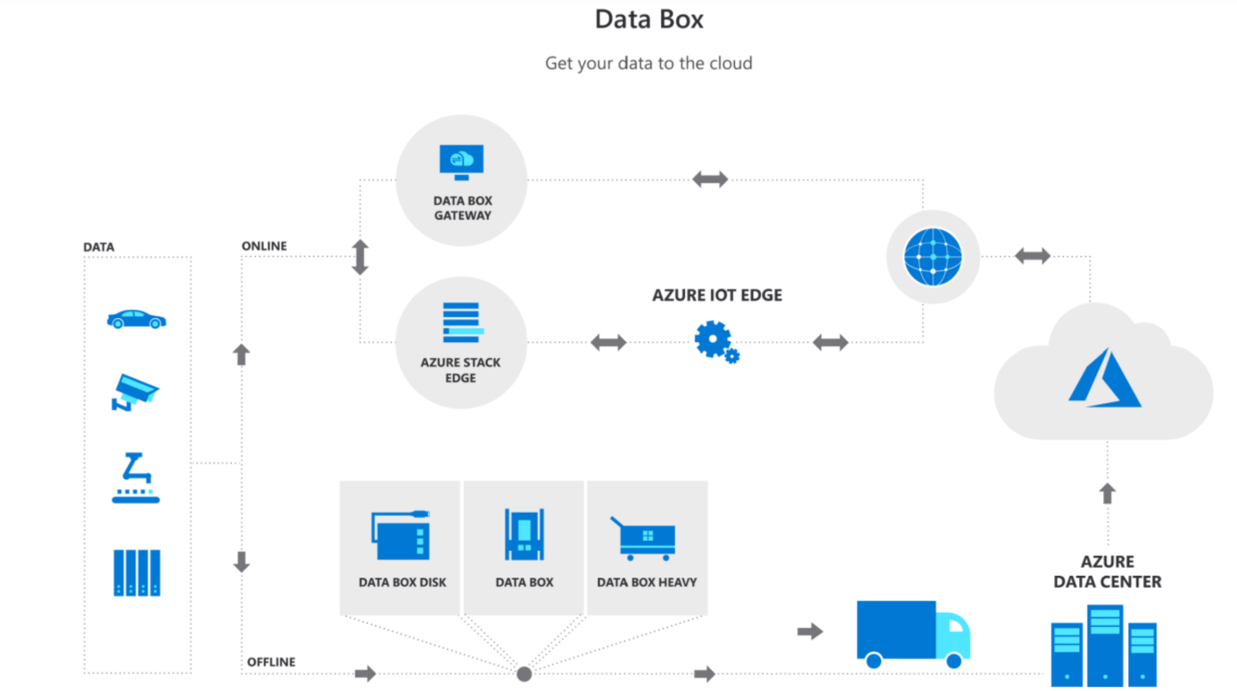
While you might have been exposed to a considerable volume of crucial insights about input/output (IO), it's worth noting that network latency could pose challenges during data loading and migrations. The size of the Oracle database itself plays a pivotal role in the achievement of a successful migration endeavor.

The subsequent script is employed to determine the dimensions of the database, discern the rate of redo generation, evaluate backups, and assess archive logs. All of these factors collectively govern the extent of the database that is to undergo migration.

Potential Tools for Migrating Oracle to Azure

DataBox- Limited network bandwidth for initial transfers of large Oracle data estates can be a challenge, but with Azure Data Box, customers can use one of three Data Box solutions to provide the right solution to migrate large data workloads to Azure:

- Data Box Disk
- Data Box
- Data Box Heavy



RMAN- Oracle's Recovery Manager is the go-to for Oracle DBAs to backup, recover, and clone databases. This is a comfortable solution for most DBAs, but consideration must be taken that RMAN is a streaming technology that can put heavy IO demands on the network and virtual machines.

Oracle Data Guard With or Without Goldengate – Oracle Data Guard, along with the standard disaster recovery solution for Oracle on Azure, is also a great way to migrate Oracle databases to Azure. With a far sync solution ensuring the changes to the standby running in Azure, a switchover to Azure, making the primary then running in the cloud can be a simple solution to a migration. If a delayed switchover is required, Goldengate can be used in conjunction with a Data Guard environment to simplify the synchronization of the on-premises and cloud environments over time.

Oracle Data Pump – Oracle's import and export tool is a logical backup and recovery tool, but like RMAN, is extremely IO heavy and less performant. All imports are done as inserts and without careful optimization of Data Pump scripting, along with keeping to smaller database workload usage, this tool can deter from meeting migration deadlines.

Third-Party Synchronization Products –

- [Quest Shareplex]
- [Qlik Replicate]
- [IBM InfoSphere CDC]

Azure Load Balancers – What do load balancers have to do with migration tools? These resources can often help balance out migration workloads and help them migrate more efficiently. Priority can be given to appropriate workloads, letting migrations occur in the background and not overwhelm the resources on virtual environments.

Important Architecture/Processes Related to Migration Success

Azure ExpressRoute plays a critical role in ensuring robust and consistent user performance when connecting to the Azure cloud. This significance is amplified, particularly when parts of the Oracle environment, such as the application tier, remain separate from the database during migration to Azure. While the ideal approach is to migrate all Oracle ecosystem components alongside the database, adopting ExpressRoute guarantees a more dependable, stable, and low-latency link to Azure.

Regarding nightly batch data transfers from on-premises sources, leveraging the benefits of ExpressRoute is key. It's crucial to meticulously assess these batch transfers to the migrating or migrated database to uncover opportunities for optimization. This entails transferring only essential data and considering the context that nightly batch transfers coincide with Oracle's nightly statistics collection and other maintenance tasks that heavily consume IO resources.

When it comes to reporting, a prudent strategy involves scrutinizing reports that employ overly broad queries like "SELECT *," or other coding practices that retrieve excessive data. In contrast to the efficient movement of data to the cloud during batch transfers, the retrieval of data from the cloud (egress) can contribute significantly to escalated cloud costs. Notably, some clients have managed to curtail consumption from 7TB to 20G by implementing modest yet impactful enhancements in their reporting queries.

For optimal performance, it's advisable to ensure that the application, middleware, and database tiers are co-located within the same availability zone for primary workloads. Moreover, during the Proof of Concept (POC) phase, it's essential to detect and address any latency concerns between these tiers, as these could potentially impact overall performance.

Project for Success

With the knowledge and techniques for evaluating, dimensioning, and structuring Oracle on Azure at your disposal, it becomes pivotal to outline a series of sequential actions aimed at comprehensively defining your project scope.

1. Commence by pinpointing the initial database slated for migration to the Azure platform.

2. Generate a comprehensive architectural diagram delineating the on-premises infrastructure, accompanied by an exhaustive inventory encompassing all pertinent systems.

3. Embark on an inclusive review process, augmenting both the diagram and inventory with the following elements:

a. Document the schemas resident within the database. Given that disparate IT factions might not correlate their work to the database or application titles, understanding the schema aids in bridging gaps.

b. Compile an exhaustive roster of all "modules" interfacing with the database. These encompass applications and executable components that could potentially elude detection unless explicitly inventoried.

c. Enumerate the assortment of maintenance routines and backup protocols.

d. Integrate into the inventory the catalog of nocturnal batch operations and other data ingestion procedures.

e. Ascertain the Recovery Time Objective (RTO) and Recovery Point Objective (RPO) mandated by the database and application. This pivotal determination often shapes the choice of storage solution within an Oracle on Azure Infrastructure as a Service (IaaS) environment.

4. Amass foundational data from Oracle, harnessing the Automatic Workload Repository to capture peak periods within the database. This process paints a lucid picture of execution durations for prevalent SQL queries and procedures, shedding light on typical workload patterns as well.

By following these structured steps, you lay the groundwork for a comprehensive and well-informed approach to orchestrating Oracle's presence on the Azure platform.

Building a Proof of Concept

The Proof of Concept (POC) needs to be strategically crafted to confirm the team's ability to effectively operate their system within a cloud environment, while also showcasing their capability to execute migrations using the resources at their disposal. While technical challenges do arise, the more common stumbling blocks tend to revolve around limited familiarity with cloud services or constrained timeframes for POC completion. Ensuring a prosperous POC outcome and deriving optimal value from it entails the following steps:

1. Develop a catalog of the ten most critical elements that demand testing, as these will substantiate the POC's triumph.

2. Employ a genuine workload for testing purposes. Contrived workloads usually yield marginal benefits, except when conducting straightforward benchmark evaluations with recommended tools.

3. Opt for a mid-range database to be utilized in the POC. Selecting the largest database, replete with intricacies, demands extensive time and resources that might impede successful outcomes.

4. Curate a POC setup with minimal complexity and reduced application connections, yet it should encompass a blend of functionalities gleaned from the team's top-ten priority list.

By adhering to these guidelines, you'll be better poised to navigate the POC process successfully, addressing potential challenges effectively and capitalizing on its potential benefits.

Switchover Best Practices

After completing the Proof of Concept (POC), it becomes imperative to give paramount importance to ensuring a seamless transition to the public cloud. Above all, a comprehensive understanding of the criticality of both the application and the database, coupled with estimating the expected downtime for the transition to the cloud environment, will significantly influence the approach taken for migration. While moving the application tier to an Azure virtual machine presents relatively straightforward possibilities, the migration of the database is often more intricate due to substantial block-level changes that might not align with available Azure services or migration tools. The Migration section proposes a range of recommended tools, spanning from employing a recovered Azure-hosted database synchronized via Goldengate leading up to the switchover, to strategies involving a Data Guard secondary that remains active until the transition takes place – a practice that is commonly followed.

The prerequisites for a successful switchover encompass the following:

1. Downtime Constraints: If a database is required to maintain continuous 24/7 uptime, the conventional outage window used for cloud migration might not be feasible. In such cases, alternatives like leveraging Oracle Data Guard for failover, implementing data replication from on-premises to Azure via change data capture, or similar strategies should be considered.

2. Nighttime Data Loads: In scenarios where nightly data loads constitute the daily data handling process, the option of directing data flows to both on-premises and Azure cloud environments could be explored. This allows for synchronization between the two environments. However, it's important to recognize the distinctiveness of each environment and to identify opportunities where existing resources can be maximized to achieve migration success.

3. Change Control Management: Embracing a change control management tool is crucial. It's not only about tracking code changes but also about effectively monitoring and managing data changes. Often, the database is the final component to be integrated into the change management process.

Upon completion of the POC, prioritizing a smooth transition to the public cloud emerges as a pivotal step. Central to this endeavor is a comprehensive grasp of the application and database's significance, along with an estimation of the downtime for the cloud migration. These factors play a pivotal role in shaping the migration strategy. While shifting the application tier to an Azure VM is straightforward, the database migration is complex due to considerable block-level changes. The Migration section recommends a spectrum of tools, ranging from employing synchronized Azure-hosted databases using Goldengate to utilizing Data Guard secondaries until the switchover, a prevalent practice.

Key prerequisites for a successful transition include:

1. Downtime Considerations: Continuous uptime databases can't rely on typical outage windows. Solutions involve Oracle Data Guard failover or replicating on-premises data to Azure through change data capture.

2. Nightly Data Loads: For nightly data operations, data could flow to both on-premises and Azure. However, unique environment attributes should guide resource optimization for a successful migration.

3. Change Control Management: Effective change management encompasses database changes, not just code changes. The database should be integrated into the change control process.

By employing a Red Hat (RHEL) or Oracle Linux image sourced from the Azure marketplace, tailored to match the customer's specific release requirements, it becomes feasible to construct Oracle software installations within an image. This image can subsequently be integrated into an Image Gallery, offering a reusable asset that eradicates deployment discrepancies and the need for manual interventions.

Upon creation, this image stands ready for integration into the Azure Resource Manager template. As an integral component of the application framework, its deployment streamlines the process for generating multiple instances and applying updates.

Inspecting Oracle on Azure Performance

Every Database Administrator is familiar with the refrain, "Nothing's changed." However, when a migration to the cloud takes place, such as moving a database to the Azure cloud, there can be significant shifts in performance due to various factors within the existing database:

1. Oracle optimizer settings.

2. Oracle statistics and management plan configurations.

3. Parameter settings.

4. The presence of Oracle bugs in the cloned database that were absent in the original.

While these are just a few examples, it's essential never to assume uniformity. For Oracle databases on Infrastructure as a Service (IaaS), regardless of their operational location, it is imperative to thoroughly investigate both the database and the underlying infrastructure.

In cases where a drop in performance occurs post-migration to Azure IaaS from on-premises, the following steps should be taken to identify the root causes:

1. Document the VM SKU in detail, considering not only the number of vCPUs and memory but also the precise SKU on which the database is running.

a. If the VM is not on a #ds VM variant, which supports premium SSDs for OS Disk and local ephemeral storage for the swap file, be aware of the performance implications. The subsequent steps will advocate for this configuration.

2. Validate that the OS disk is residing on premium SSD storage, ideally P6-P10 disks.

3. Confirm that the Linux swapfile is situated on the local ephemeral storage of the VM.

4. Record details of each attached storage, including the storage type, size, and any active caching for each disk. Verify the location of datafiles and logfiles on each disk.

Once the infrastructure aligns with recommended practices for optimal performance, attention should shift to optimizing the database workload. Utilizing tools such as the Automatic Workload Repository

(AWR) and other Oracle tuning products can aid in this endeavor. Key recommendations for Oracle optimization include:

1. Analyze AWR's "Top SQL by Elapsed Time" between the original on-premises and current Azure setups for comparable workloads.

   a. Identify SQL statements displaying performance degradation.

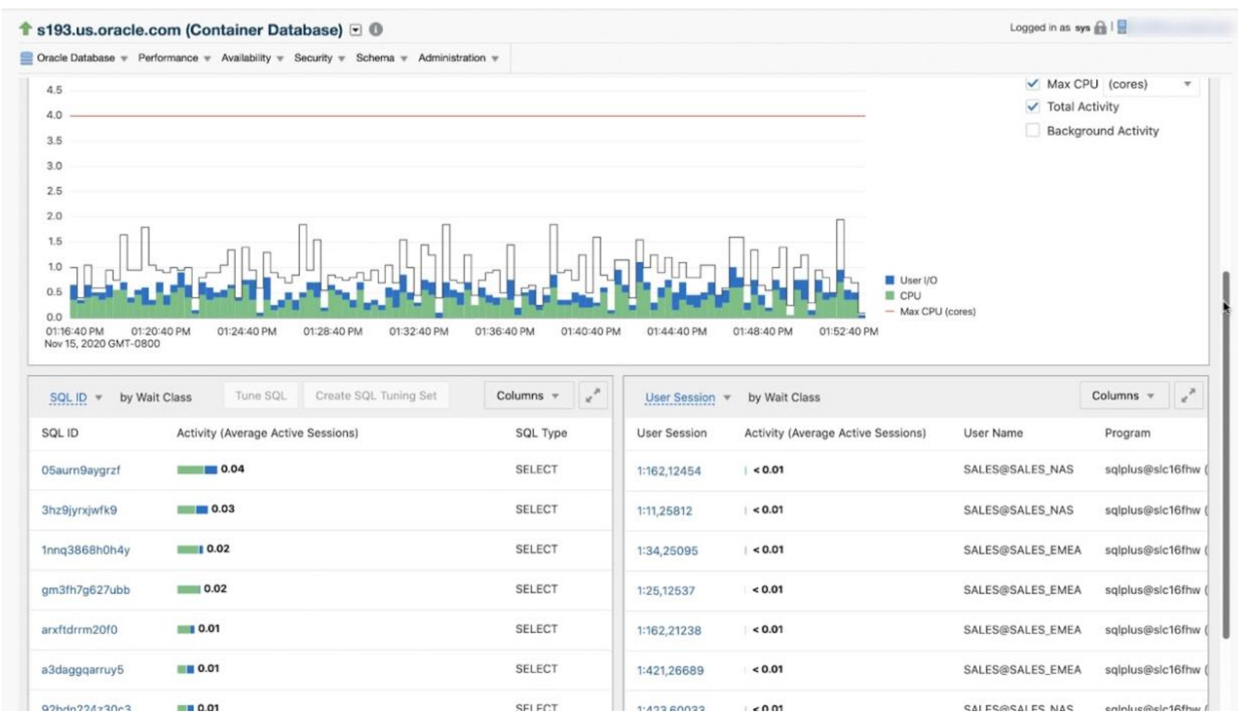
   b. Compare execution times of the same SQL in both reports to assess degradation.

   c. Scrutinize top foreground and background performance metrics for signs of deterioration.

   d. Recognize that high IO maintenance and backup tasks could lead to workload throttling. Verify that excessive non-user tasks are not monopolizing IO resources.

   e. If an upgrade accompanied the migration, ensure that increased resource usage and outdated parameters are not adversely affecting performance.

The Oracle Specialists from the Cloud Architecture and Engineering Team commonly advise allowing Oracle to operate in Azure for a span of 2-6 months before embarking on cost optimization efforts. This strategy involves investigating and testing reduced resource utilization in the infrastructure to save costs within the cloud environment. Testing should be confined to a sandbox environment, using AWR reports in tandem with infrastructure data. The same guidelines for "Inspecting Performance for Oracle on Azure" apply, with the focus on identifying areas of satisfactory performance through Oracle Cloud Control (Enterprise Manager) and Azure Monitor, which can indicate underutilization of resources.



After verification, contemplate adopting a lower service tier to attain acceptable performance without compromising cost-efficiency. It is advised to approach this process gradually, spanning an extended timeframe, and to implement alterations incrementally. Utilize all the discussed tools to closely monitor any potential decline in performance during each adjustment.