Using an AWR Report to Size an Azure VM

Authors: Kellyn Gorman and Tim Gorman

Scope

This document uses the Automatic Workload Repository (AWR) to gather workload data for an Oracle database and provide initial estimates for sizing instance types and storage in Azure, once specific fields from the AWR report are populated to an Excel spreadsheet.

Disclaimer: Each version and database type of the 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 immensely.

Assumptions

- AWR Report with workload report captured during a *peak workload* period of time
- 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

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.

If you need to know how to run an AWR report, the steps are as follows:

The DBA should log into the database server as a privileged user and run the following from a location they can write a file to:

@$ORACLE_HOME/rdbms/admin/awrrpt.sql;

For a RAC environment, run the "RAC aware" report:

@$ORACLE_HOME/rdbms/admin/awrgrpt.sql;

- The report format: HTML

The AWR script will display snap IDs that are available for generating a report.

If the person running the report knows what time period knows a range of AWR Snap IDs representing a period of peak workload on the database, then they should enter that range of Snap IDs as the "`Begin Snap`" and "`End Snap`" parameters.

However, if the person running the AWR report does not know of a range of AWR Snap IDs representing a period of peak workload, then perhaps they can make use of the SQL script "`busiest_awr.sql`".

Running the "`busiest_awr.sql`" in the Oracle SQL*Plus utility (or any other command-line interface which supports SQL*Plus operators) can execute this query, showing the top five *busiest* 3-hour time periods currently retained in the Automatic Workload Repository tables. The query in the SQL script will query the DBA_HIST_SYSSTAT view to focus on database statistics named "`CPU used by this session`" and "`physical reads`", representing *CPU utilization* and *I/O operations*, respectively. Generate a report using one of the suggested AWR Snap ID ranges as the "`Begin Snap`" and "`End Snap`" parameters.

The AWR report should be uploaded from the database server and then emailed to those performing the review.

The AWR Worksheet

The first three columns:

DB Name: the unique name given to the database.

Database		
Id	**Name**	**RAC**
1633071752	GBACSPRP	YES

Instance Name: Is the same as individual database node names in RAC or often the same as the DB Name for non-RAC databases.

Host Name: The name of the host. For RAC, each node will have a unique name.

Instance	Host
gbacsprs1	gbslo1exdb01.dunnhumby.c

In a non-RAC environment and above, this information, along with information you will need for later values, like CPUs, memory, etc. are presented at the very top of the report:

DB Name	DB Id	Unique Name	Role	Edition	Release	RAC	CDB
ACDB10P	2381149518	acdb10p	PRIMARY	EE	12.2.0.1.0	NO	NO

Instance	Inst Num	Startup Time
acdb10p	1	16-Jan-20 07:01

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
ac-db10	Linux x86 64-bit	128	64	4	2015.90

Elapsed Time and DB Time: These two sections are commonly next to each other throughout the report.

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

Or

Elapsed:	1,199.41 (mins)
DB Time:	33,968.17 (mins)

DB CPUs: This can be a confusing metric, as CPU data is in numerous fields, but the value we're searching for is referred to as "DB CPU(s)". Enter it for each instance involved in the estimate. DB CPU is a larger value than much of the other values displayed in the overall report, as it's displayed in seconds. How many seconds was a CPU was used or you waited on a CPU:

Statistic Name	Time (s)	% of DB Time	% of Total CPU Time
sql execute elapsed time	1,190,253.99	58.40	
DB CPU	955,928.53	46.90	71.63

Foreground Wait Classes

I#	User I/O(s)	Sys I/O(s)	Other(s)	Applic (s)	Commit (s)	Network (s)	Concurcy (s)	Config (s)	Cluster (s)	DB CPU (s)	DB time
1	3,810.80	18,629.87	4,703.05	48,118.02	922.14	2,417.59	482.03	1.76	1,772.12	189,371.91	308,469.93
2	2,530.80	115.61	2,353.61	58,401.86	1,003.08	3,905.13	496.52	2.11	1,665.19	209,455.96	283,219.56
3	3,774.36	1,560.68	3,795.82	50,500.30	1,011.93	2,760.93	388.57	3.23	1,339.03	217,165.35	286,029.28
Sum	10,115.96	20,306.16	10,852.48	157,020.18	2,937.16	9,083.65	1,367.12	7.10	4,776.34	615,993.22	877,718.77
Avg	3,371.99	6,768.72	3,617.49	52,340.06	979.05	3,027.88	455.71	2.37	1,592.11	205,331.07	292,572.92
Std	728.72	10,297.44	1,184.83	5,383.11	49.49	778.87	58.59	0.77	225.60	14,348.51	13,838.71

CPUs/Cores: Hyper-threading makes it important to have both these numbers. We commonly calculate off of the Cores value and ensure that you update the CPU calculation for it in the spreadsheet if you do note that there is hyperthreading involved. For the example below, a 3-node RAC has 320 hyperthreaded CPUs, with 160 CPU cores total for each.

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

OR

CPUs	Cores
128	64

Memory(GB): Memory is captured in the same line as CPU information, but it is calculated differently than we need in our spreadsheet. Remember to convert from MB to GB as part the steps when you enter the info.

Memory (M)
6,191,158.41

/ 1024= Correct Value for Spreadsheet

OR in newer versions, displayed in GB:

Memory (GB)
2015.90

%Busy CPU: This value is clearly stated in the report and is used by the formulas in the spreadsheet to indicate possible CPU saturation. Oracle marks database sessions as either "ON CPU" or waiting for something, but to know if sufficient CPU was available during observation is part of our estimates. This is another value that can be confusing to gather. Go to 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 vale. 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	

In newer versions of the AWR, the host memory, SGA and PGA is displayed more clearly and in one section:

Memory Statistics

	Begin	End
Host Mem (MB):	2,064,280.1	2,064,280.1
SGA use (MB):	1,331,200.0	1,331,200.0
PGA use (MB):	5,447.9	6,495.5
% Host Mem used for SGA+PGA:	64.75	64.80

Again, ensure you take the larger of the begin/end values for each.

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 File Type (per Second)

- Total Reads includes all Filetypes: Data File, Temp File, Archive Log, Backups, Control File, Data Pump Dump File, Flashback Log, Log File, Other, etc
- Total Writes includes all Filetypes: Data File, Temp File, Log File, Archive Log, Backup, Control File, Data Pump Dump File, Flashback Log, Log File, Other, etc

I#	Reads MB/sec Total	Data File	Temp File	Writes MB/sec Total	Data File	Temp File	Log File	Reads requests/sec Total	Data File	Temp File	Writes requests/sec Total	Data File	Temp File	Log File
1	46.94	44.57	0.01	1.51	0.02	0.01	0.01	139.59	45.70	0.26	10.37	1.33	0.17	5.53
2	34.42	34.30	0.01	0.08	0.02	0.01	0.01	41.86	36.75	0.22	8.96	1.26	0.18	5.92
3	60.35	57.89	0.01	0.09	0.02	0.01	0.02	160.81	60.62	0.32	9.85	1.41	0.24	6.50
Sum	141.71	136.76	0.03	1.67	0.05	0.03	0.04	342.25	143.08	0.80	29.18	3.99	0.59	17.95
Avg	47.24	45.59	0.01	0.56	0.02	0.01	0.01	114.08	47.69	0.27	9.73	1.33	0.20	5.98

Read IOPs/Write IOPs: Like throughput, this section can be displayed in different parts of the AWR report, but often is in the Load Profile towards the top of the report or in the IO Statistics in the mid-section of others.

System Statistics - Per Second

I#	Logical Reads/s	Physical Reads/s	Physical Writes/s	Redo Size (k)/s	Block Changes/s	User Calls/s	Execs/s	Parses/s	Logons/s	Txns/s
1	20,652.58	5,540.18	3.26	12.25	176.16	19.10	22.49	10.40	0.92	2.00
2	28,469.81	4,392.11	3.29	12.03	222.90	11.57	33.02	9.98	0.98	2.26
3	33,854.74	7,411.34	3.66	13.47	251.38	13.87	27.95	11.56	0.97	2.51
Sum	82,977.14	17,343.63	10.20	37.75	650.44	44.54	83.46	31.94	2.87	6.76
Avg	27,659.05	5,781.21	3.40	12.58	216.81	14.85	27.82	10.65	0.96	2.25
Std	6,638.31	1,523.98	0.22	0.77	37.97	3.86	5.27	0.82	0.03	0.25

As expected, newer versions of the single instance report will group all the IO data in the Load Profile section and make it simpler to read:

Load Profile

	Per Second	Per Transaction
DB Time(s):	28.3	0.1
DB CPU(s):	13.3	0.1
Background CPU(s):	5.3	0.0
Redo size (bytes):	35,799,937.3	155,727.0
Logical read (blocks):	1,066,800.1	4,640.5
Block changes:	123,113.1	535.5
Physical read (blocks):	6,142.4	26.7
Physical write (blocks):	4,092.1	17.8
Read IO requests:	2,048.5	8.9
Write IO requests:	1,059.1	4.6
Read IO (MB):	96.0	0.4
Write IO (MB):	63.9	0.3

Extrapolation 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 instances:

Peak CPU factor	3.00
Est'd RAM factor	1.50
vCPU HT factor	2.00
%Busy CPU threshold	0.75
%Busy CPU multiplier	1.25
IO metrics (IOPS & MB/s) fudge factor	3.00

These fields are intended to extrapolate the observed AWR workload higher than the captured workload levels. For example, if the captured AWR metrics actually represented the highest peak workload on-prem, then there might be no need for extrapolating higher for sizing on Azure. However, it is generally wise to extrapolate a somewhat higher, to ensure that new environment in Azure can safely accommodate the peak workload. In this case, setting "`Peak CPU Factor`" to a value like 1.50, which will multiply the observed values only by a conservative 150%, is a good idea.

However, suppose that the on-prem environment is an Oracle Exadata engineered system? In that case, there might be a need for much more CPU power in the Azure virtual machine than observed in AWR? In that case, setting "`Peak CPU Factor`" to a value like 4.00 or 6.00, which will multiply the observed values by 400% to 600% higher, might be reasonable.

This might also be true if we are not sure that the metrics captured in the AWR report came from a peak period, so we might need to compensate using extrapolation what we missed in actual observations.

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

Peak CPU Factor: 1.50 is standard, 3.00 for an on-prem workload of which we're not certain is peak, and 4.00 to 6.00 for a workload that might have a larger expectation of variance once it goes to the cloud.

Est'd RAM Factor: Same for CPU, but for RAM estimate. Normal is 1.50 or 2.00, but 4.00 might be normal for an Exadata where the SGA is commonly shrunk to promote offloading.

vCPU HT Factor: Commonly 2.00 and this should be the default going to IaaS Azure VMs. The only change that could ever be made is if the Azure vCPUs are not hyperthreaded, in which case this value should be "1.00"

Busy CPU threshold: 0.75 is the default, which indicates a threshold of 75% observed CPU utilization. When CPU utilization is higher on a Linux-based server, then there is likelihood that some of the time that Oracle has marked as "ON CPU" is actually performing CPU scheduling activities, and is not performing work (a.k.a. processing). In this case, we may want to extrapolate the CPU utilization even higher than what is indicated in **Peak CPU Factor**, leading us to the next variable...

%Busy CPU multiplier: Default value is 1.25, and is used to further increase the estimated number of Azure vCPUs by 125% when the "%busy CPU" observation from AWR exceeds the "Busy CPU Threshold" variable described above. Again, it is all about the fact that Oracle may have designated "ON CPU" to mean "processing work" when in fact the CPUs might be "thrashing" due to high utilization.

IO metrics(IOPS & MB/s) fudge factor: A value of 2.00 (or 3.00) is for OLTP/transactional database applications, or please consider 3.00 or 4.00 for DSS/OLAP database applications, and please consider 6.00 when the source database(s) are on an Exadata environment.

Important note about *extrapolation factors*

Everything in the upper portion of the "AWR" worksheet of the spreadsheet is based on observed facts from the source AWR report.

If we are certain that the source AWR report captured peak workload, then we should only need lower, more-conservative values in the *extrapolation factors*. On the other hand, if we have any reason to believe that the source AWR report does not reflect the highest peak workloads on-prem, then we should use higher, more-corrective values in the *extrapolation factor* fields.
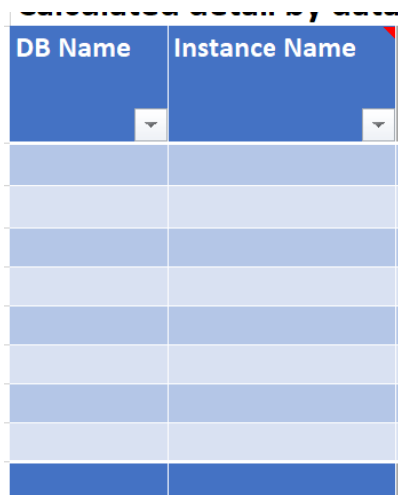
These *extrapolating factors* put the *art* back into the *science*. Please be careful with your adjustments to these calculating factors, and please be ready to reasonably justify your changes. They are not intended to be used to tweak the calculations to preferred values. Rather, they are intended re-introduce some of the reality that might have been missed in the source AWR report.

Please be cautious!

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 DBName and Instance Name, duplicating the DB name if necessary, that corresponds to the instance name. Do not leave the first column blank if you filled in the second.

DB Name	Instance Name

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.

Host	Name
Total	

3. Enter the db name, (database name, NOT instance name) in the third section, first column. If working on a RAC environment, the RAC database will be listed by the DB name one time, not for each node in this section. The Excel spreadsheet will calculate and total the resources required for a **single instance,** as this is our primary goal to achieve a fully supported environment on Azure by Oracle.

DB Name
Total

As you're entering the values into this second worksheet, calculations will appear. Once complete, you should have values for each database to size the workload into Azure. These values will then give you the information you need to choose one or more IaaS Azure VMs to size out a solution for the Oracle customer.

Example of Calculations for RAC to Single Instance

The following is an example of the output from a customer engagement. This involves two databases, both 2-node RAC environments. Notice that the DB Name column is listed twice for both, then the instance name is unique. No other information was filled in, as the values in the previous worksheet automatically populates and calculates what is needed.

Calculated detail by database instance

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
ARPH2PPD	ARPH2PPD1	3059.755%	48.951%	310	1,296.50	125.69	61.20
ARPH2PPD	ARPH2PPD2	2915.155%	45.965%	310	1,433.08	127.69	58.30
ARPH2PRD	ARPH2PRD1	669.554%	8.185%	310	2,538.29	190.88	13.39
ARPH2PRD	ARPH2PRD2	679.164%	8.366%	310	3,021.84	276.57	13.58
		7323.628%	111.469%	1,240	8,289.71	720.83	146.47

In the second section, only the host name was populated to the first column for each of the nodes for the RAC instances. As there are two nodes each for the two databases, four entries are added and the values populate from the first worksheet.

Aggregated calculations by host

Host Name		%DB Ti
dbsls504		
dbsls505		
dbslp2030		
dbslp2031		

In the last section, I only listed the two, global database names. The data for each of the nodes for each of the databases is calculated and total resources are displayed for the environment to be moved to Azure IaaS VMs. With the factoring numbers taken into consideration, we have average workloads from the AWR and then peak workloads which are calculated from the workloads and the factoring numbers.

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
ARPH2PPD	2987.455%	620	930	2,729.58	253.38	4,094.37	380.07	32	84
ARPH2PRD	674.359%	620	930	5,560.13	467.45	8,340.20	701.18	24	64
Total	3661.81%	1,240	1,860	8,289.71	720.83	12,434.57	1,081.25	56	148

For our examples:

ARPH2PPD will require:

- 32 vCPU for an average load and 84 vCPU for a max workload.
- A server with 930G of memory and 620G allocated to the database.
- Disk IOPS 4094 and 380MB/s throughput

Calculations can be seen for the second database to be migrated to a single instance, ARPH2PRD.

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 towards the project. The values we have here is what we require to size out the Azure VM.

Choosing an Azure instance type using the estimates

Continuing with the example of the evaluation of a database named ARPH2PPD, the spreadsheet reported that we observed 32 Azure vCPUs for the average workload collected from the AWR report. Next, the spreadsheet extrapolated 84 vCPU for a hypothetical "peak" workload, which is the metric that we'll use to size an instance type in Azure.

Here are the steps…

1. Navigate to the web page HERE on Azure instance types for Linux
2. Scroll past the pricing information, down to the section entitled **Explore all Virtual Machine options**
3. In the drop-down box entitled **OS/Software**, choose *Red Hat Enterprise Linux*
4. In the drop-down box entitled **Region**, choose the *Azure region* you prefer
5. In the drop-down box entitled **Currency**, choose the *national currency* you prefer
6. In the drop-down box entitled **Display pricing by**, choose *Month*

The resulting choices should look something like this…

Now scroll down into the E-series or M-series instance types to choose the instance type with more than enough vCPUs and vRAM to meet the estimates from the spreadsheet.

Using the example (above) of the evaluation of a database named ARPH2PPD, the spreadsheet reported an estimated 84 vCPUs for hypothetical "peak" workload, as well as an estimated 930 GiB of vRAM. Looking over the available E-series and M-series instances types in the East US region, we see…

- E96a v4 & E96as v4 (96 vCPUs and 672 GiB vRAM)
- M64 & M64s (64 vCPUs and 1024 GiB vRAM)
- M64m & M64ms (64 vCPUs and 1792 GiB vRAM)
- M128 & M128s (128 vCPUs and 2048 GiB vRAM)
- M128m & M128ms (128 vCPUs and 3892 GiB vRAM)

This narrows down the choices considerably, but there are still a lot of choices. To narrow this list further, let's examine the IOPS and I/O throughput limits on each of these instance types, to see if the requirements of the ARPH2PPD database disqualify them.

Here are the steps…

1. Navigate to the web page HERE for more information on the E-series instance types
2. There is no information for the E96a v4 or E96as v4 instance types, but there are IOPS, read throughput, and write throughput limits for the E64 instance types
   a. The limits for "IOPS, read-throughput, and wri-throughput is shown as "96000/1000/500", meaning 96000 IOPS, 1000 MB/s read throughput, and 500 MB/s write throughput
3. Navigate to the web page HERE for more information on the M-series instance types
4. The IOPS and total I/O throughput limits for the M64 instance types is shown as "40000/1000" meaning 40000 IOPS and 1000 MB/s throughput
5. The IOPS and total I/O throughput limits for the M128 instance types is shown as "80000/2000" meaning 80000 IOPS and 2000 MB/s throughput

Let's see if we can narrow the list of possible instance types further. So far, when considering just vCPU and VRAM, we have the following list…

- E96a v4 & E96as v4 (96 vCPUs and 672 GiB vRAM)
- M64 & M64s (64 vCPUs and 1024 GiB vRAM)
- M64m & M64ms (64 vCPUs and 1792 GiB vRAM)
- M128 & M128s (128 vCPUs and 2048 GiB vRAM)
- M128m & M128ms (128 vCPUs and 3892 GiB vRAM)

Now, with the spreadsheet estimating 4,094.37 IOPS and 380.07 MB/s for the ARPH2PPD database, we can see that both the E-series and M-series instance types have more than enough IOPS and throughput to manage the estimate of 4094.37 IOPS and 380.07 MB/s.

So, any of the instance types initially suggested by the vCPU/vRAM estimates are viable options for the our example ARPH2PPD database. There are other considerations, such as write acceleration (available only with M-series) and cost (E-series is less expensive than M-series), and we won't be covering those considerations in this paper, so we'll just stop right here.