Oracle Migration Accelerator - Data Collection

Introduction

If you're considering migrating your Oracle DB to Microsoft Azure, it's crucial to evaluate the technical feasibility, potential for rightsizing, and cost and effort estimates. This evaluation is an essential and mandatory part of the Microsoft Oracle Migration Accelerator Program. To begin analyzing your Oracle Database estate, the first step is generating an AWR (Automatic Workload Repository) report. This detailed performance analysis report is automatically generated by Oracle databases and provides comprehensive information about database performance, wait events, system resources, and much more. The AWR report is included with the Oracle Database Enterprise Edition and the Diagnostic Pack option. If you're running an Enterprise edition without a Diagnostic Pack license or Standard Edition, you can still use a Statspack report level 7 for the same purpose.

Preamble

When sizing resources for Oracle database workloads in Azure, it's critical to measure the actual CPU, memory, and I/O utilization of the database workload in its current environment, whether on-premises or in another cloud. It's essential to distinguish between the actual database workload and the existing on-premises host configuration.

Migrating database workloads to Azure requires a fresh approach, rather than replicating the on-premises host configuration. Discussions about hardware, physical hosts, or virtual machines are less relevant in the cloud context.

On-premises Oracle database environments are often over- or under-provisioned due to the typical 3-5 year refresh cycle for hardware. At the beginning of a cycle, workloads may be over-provisioned due to hardware purchased with expectations for the next five years. Toward the end of the cycle, some workloads may be under-provisioned due to estimates made five years ago falling short of reality.

To optimize the customer experience, resources are "right-sized" based on observed utilization, with no guesswork or assumptions. Unlike on-premises environments, cloud resources are regularly updated, and resources can be resized as needed at any convenient time, eliminating the need for refresh cycles.

In order to assess database workload, we request a standard Oracle *Automatic Workload Repository* (**AWR**) report from periods of peak workload information stored within the built-in archives of the customer's production database. From the thousands of metrics displayed in these AWR reports, we focus on a few to determine *actual CPU utilization*, *actual memory (RAM) utilization*, and *actual I/O utilization*. These metrics are most useful if the AWR reports cover a time period when ***peak workload*** was experienced, because peak workload is the "worst-case scenario" for resource usage and therefore contains the best information to determine how much and what type of resources to provision in Azure. Once we understand actual utilization for CPU, RAM, and I/O, then it becomes easy to choose the appropriate Azure resources for compute and storage.

AWR is enabled only on Oracle Database Enterprise Edition, and only with the additional Oracle Enterprise Manager Diagnostics pack licensing, so there are many situations (including the use of Oracle Database Standard Edition) where AWR is not available. In that case, we recommend installing and generating reports from the open-source Oracle STATSPACK utility – see script here busiest_statspack.sql, which is a predecessor package of AWR and is available when AWR is not. There are only slight differences between the two packages, and STATSPACK runs on any version or edition of Oracle database.

The exact timeline of peak workloads may be uncertain, therefore we usually request two types of AWR reports one during mostly during the peak workloads in general several hours and one over a longer time period to get OLAP related queries over about **a week** (i.e. 7 days), which generally captures peak workloads and volume based ones at some point. For non-RAC databases, a single standard AWR report covering the week is sufficient. For RAC databases, either a single standard AWR report for each RAC database instance is sufficient, or a single global RAC AWR report encompassing all database instances is sufficient.

Additional, more exact and accurate peak workload information, can be acquired by downloading the "`busiest_awr.sql`" script from Github HERE or via ATROPOSS run it once using Oracle SQL*Plus for each database, and then review the output which will display the top five (5) *busiest* AWR "snapshots" in the AWR repository in the database.

In addition to the AWR reports, which do not contain any information about database size, volume of generated transactions (redo), or backups, we also request you download the "DBspace.sql" script from Github HERE (standalone script not integrated into ATROPOSS yet) and the "`dbSizeCSV.sql`", run it once using Oracle SQL*Plus for each database, and then return the spooled text file via email.

If databases are deployed on virtualized environments – ESX the static RVTool can be executred as well to get additional information about the physical environments like cpu family, memory, consumed cpu and memory. Please keep in mind that the rvtool is static and should have a the same time periods like the AWR's or at least during the same time period. From the output of the following RVTool we need the following 2 csv's (RVTools_tabvHost.csv, RVTools_tabvInfo.csv), stored in a separate CSV and uploaded in the same step like the AWR's, statspack, and dbSiveCSV.csv.

Information have to use the RVTool is available here PDF short description to RVTools english ver.

How To

Verify that AWR can be generated

The following should be done by the customer, to ensure that an AWR can be generated. Otherwise, a Statspack report should be generated.
- Log on to the database in question using an account with admin privileges
- Run 'show parameter control_management_pack_access'
- Verify that parameter control_management_pack_access is not set to NONE. Expected value is either 'DIAGNOSTIC' or 'DIAGNOSTIC+TUNING'.

General Guidelines

To generate comprehensive AWR reports for a database right- sizing exercise, follow these guidelines:
- Generate the busiest AWR report for peak workload during short periods, such as one hour or thirty minutes.
- Generate a batch/nighttime AWR report to cover non-daily activities such as batch jobs, backups for an extended period, like 6-7 hours.
- If available, provide any additional AWR reports that may cover important characteristics of the database workload, such as end-of-month or end-of-quarter reporting periods.

Get busiest AWRs

To determine the peak load times the customer should do the following.
- Get the *busiest_awr.sql* script.
- If this is not a multi-tenant database, do the following:
    - o Log on to the database in question using an account with admin privileges.
    - o Run the busiest_awr.sql script, to determine the five busiest snap ids, note the top three ones down for future reference.
- If the database in question is a multi-tenant, do the following:
    - o Log on to CDB$root using an account with admin privileges.
    - o Run the busiest_awr.sql script, to determine the five busiest snap ids, note top three ones down for future reference.

Generate AWR

The following steps should be taken by the customer to generate the AWR reports.

A. If this is not a multi-tenant or RAC database, do the following:

1. Log on to the database in question using an account with admin privileges.
2. Run these commands:
   - @$ORACLE_HOME/rdbms/admin/awrrpt.sql
   - When prompted choose 'html' as the output format
   - Press return to get all snapshots.
3. When prompted, indicate a relevant snapshot id as previously identified (peak load, batch/nighttime/other relevant) as the begin value. Increment the snapshot id as the end value according to your AWR report coverage.

For example, when generating the busiest AWR, increment the begin snapshot id by one as the end value.

▪ Accept default values for the output file.

▪ Specify a name to your AWR report to be generated or press return for default value.

Note that the AWR report will be created in the current working directory.

B. If the database in question is a multi-tenant, do the following:

1. Log on to CDB$root using an account with admin privileges.
2. Run these commands
   - @$ORACLE_HOME/rdbms/admin/awrrpt.sql
   - When prompted choose 'html' as the output format
   - Press return to get all snapshots.
3. ▪When prompted, indicate a relevant snapshot id as previously identified (peak load, batch/nighttime/other relevant) as the begin value. Increment the snapshot id as the end value according to your AWR report coverage.

For example, when generating the busiest AWR, increment the begin snapshot id by one as the end value.

- Accept default values for the output file.
- Specify a name to your AWR report to be generated or press return for default value.

Note that the AWR report will be created in the current working directory.

C. For RAC database, depending on whether it's multi-tenancy or not, repeat the above on each node in the RAC

Executive Summary

To summarize what is requested from the customer's Oracle DBA team to begin sizing for Azure…

1. **AWR (or STATSPACK) report(s)**
   1. From the most recent week by default, or from any recent peak workload periods, preferably?
2. **Output from DBspace.sql and `dbSizeCSVNew.sql` script**
   1. Download Oracle SQL*Plus DBspace script from Github HERE – standalone script including backup/archiving and the dbSizeCSV.sql execute it in SQL*Plus while connected as SYSDBA, then share output via email
   2. If customer prefers, instead of downloading and running the script, then simple answers to these questions are sufficient…
      - What is the size of database in GB
      - What is the average daily volume of transaction logs in GB
      - What is the expected annual growth rate of database

3. Executing the script dba_feature_usage.sql dba_feature_usage.sql and store the output in a file
4. **(optional) Integration ESX information of virtualized deployed databases**
   - Create the two csv files RVTools_tabvHost.csv, RVTools_tabvInfo.csv) our of the output of the rvtools.
5. **(*optional*) Recovery Time Objective (RTO) requirement**
   - Expectations, expressed in seconds, minutes, hours, or days, for full return to service of the database after DR failover
   - This is *optional* because we can assume an RTO of 4 hours as a starting point, and then confirm at a later time?
6. **(*optional*) Recovery Point Objective (RPO) requirement**
   - This is *optional* because we'll assume RPO=0, assuming that *any* data loss during DR failover cannot be tolerated

Obtaining a standard Oracle AWR report takes only seconds to generate, and can be obtained through several methods…

- SQL Developer v4.x and up
- Oracle Enterprise Manager 11g, 12c, and 13c
- SQL*Plus directly from the database server
  - Log in as `oracle` account on the database server host
  - Change directory to `$ORACLE_HOME/rdbms/admin`
  - If the database is non-RAC, then please run the command `sqlplus / as sysdba @awrrpt` to generate a single-instance report
  - If the database is RAC, then please run the command `sqlplus / as sysdba @awrgrpt` to generate a RAC cluster report

Provided with the information requested here, we will make accurate recommendations for compute, storage, backup, and disaster resiliency (DR) resources in Azure for the production Oracle database workload in a matter of hours, usually with same day turnaround.

FAQs:

Why AWR is required?

AWR reports are instrumental in our process, as they provide critical insights into the database workload, enabling us to make accurate assessments of the appropriate size for our Virtual Machines (VMs). In essence, AWR reports deliver a comprehensive analysis of database performance over a specific period of time. These reports capture, process, and maintain performance statistics for problem detection and self-tuning purposes. The data captured provides valuable statistics on compute, memory, and storage usage, as well as the load on the database, all of which are crucial in helping us to make informed decisions about the right size of VMs that we need to employ.

Why dbSizeCSVNew is required?

Azure is a data estate platform with various different data patterns. The output of the script is uses to enrich the reverse engineering of the data platform and to weight the resulted entities for recommendation of the appropriate data service. Next to a lift & shift or modernization in direction to Postgresql, has Azure multiple additional services available like ADX, Databricks, SQL, Redis, etc.!

Why DBspace is required?

When planning a migration (https://github.com/Azure/Oracle-Workloads-for-Azure/blob/main/az-oracle-sizing/dbspace.sql) of a database or any kind of workload to a new environment, understanding the database size (DBspace) and disk consumption is critical for several reasons:

Capacity Planning: The total size of the database and its growth rate are essential to correctly size the storage in the new environment. You don't want to run out of space in the middle of the migration or soon after.

Cost Estimation: Cloud services usually have a pay-as-you-go model. Knowing how much storage you need, and the speed of your I/O operations, can help you choose the right type and size of storage, which directly impacts the cost.

Do I need to share output of *busiest_AWR.sql, busiest_statspack.sql* with Microsoft?

Output from this script is intended for the use of the customer, and does not need to be shared with Microsoft.