ALF DB Sizing

Introduction

ALF provides a framework of best practices and advises for Hybris partners and customers. Long term goal in ALF is to provide also capacity planning and sizing information for Hybris installations. Purpose of this document is to define factors that typically have effect on sizing on Oracle database to help in estimation of needed hardware resources. This document provides a study of factors that affect to Oracle performance on Hybris installations.

This page was written as part of this DB sizing exercise Oracle XE Docker on Windows and Docker was used to create a test system on local laptop.

Model for sizing database

Database sizing is approached in this section from following aspects that are generally used for software performance estimation:

- CPU utilization
- Disk usage
- Disk speed
- Memory

Requirements

Hybris 6.0 and Apparel store, Electronix Store and Powertools store are used for this estimation.
Together they have 7656 products in staged and online catalogs

apparel: Online 1935, Staged 1935
electronic: Online 267, Staged 267
powertools: Online 1359 , Staged 1359

For any sizing estimation, an input is required to analyse requirements. For database sizing where there is no production data available, requirements have to be calculated from different scenarios. In this chapter are presented few typical Hybris scenarios.

- B2B scenario; 500 concurrent users
  - Products 1359
  - Product catalogues : 1
  - Conversion rate; up to 0,5%-2,5% (B2C) avg. 2%
  - Page views per visitor; 5
  - Order lines; 2
  - Session duration; 15 min
  - Busiest hours; 15%
  - Batch ratio is 1000:1
  - Batch processing is executed when possible (no good time for data processing as site is available 24/7)
  - Business hours; 16
  - Total users per day: (60/15)* 500*16 = 32000
  - Page views per day: 32000*5=160000
  - Pages view per hour: 10000
  - Page views per second: 2,7

From these scenarios, we can calculate that demand for storefront is 2,7 PI/s (Page Impressions / second).
When PI/s is known, with few additional attributes like conversion rate and batch ratio and amount of products in different catalogues, it is possible very roughly to estimate how many queries and transactions per second (TPS) the system executes against the database. Oracle defines transaction to be each committed or roll backed transaction. Queries are not transactional in Oracle.

As database is shared service, different type of workloads should be measured and categorized in Hybris we to have understanding how they utilize database. These workload categories are for example:

- Browse assortment online
- Add to cart and checkout
- Catalog synchronization
- Catalog import
- Search index update

All of the above mentioned workload categories have different usage profile. For catalog synchronization, most of operations are

INSERT/UPDATE. For cart and checkout, the profile is mix of selects and updates. Depending if modules like BTG is used, browsing assortment is either pure selects or mix of selects and insert/updates.

For example, for online browsing, if one page view would generate 0,5 transactions and 2 queries (most done by cache) we would get 1,5 transaction per second and 6 queries per second on database.

In this document, only catalog import and catalog synchronization is analysed further as getting enough concurrent users would require a performance test script and infrastructure to be prepared.

Database size for Hybris

Following result is returned for when querying sizes from sys views (filtered with 15 largest tables):

Table name	Size (MB)	KB per ROW	ROWS
PRODUCTFEATURES	26	0,323405082	82324
ATTRIBUTEDESCRIPTORSLP	21	0,067417421	318968
ITEMSYNCTIMESTAMPS	10	0,15508807	66027
PRODUCTSLP	9	0,76513076	12045
ATTRIBUTEDESCRIPTORS	8	0,304932068	26865
MEDIAS	8	0,326166587	25116
STOCKLEVELS	8	0,117736673	69579
CUSTOMERREVIEWS	6	2,203730273	2788
CAT2PRODREL	3	0,155828345	19714
PRODUCTS	3	0,431339511	7122
PROPS	3	3,386990077	907
COMPOSEDTYPESLP	2	0,180934712	11319
PRODUCTTAXCODES	2	0,130396027	15706
PRICEROWS	0,8125	0,147439305	5643
CMSCOMPONENT	0,625	0,237037037	2700

Average size per one row stored in tables is 0,385 KB.
Hybris database size with this setup is 145 MB.

Total database size for Oracle estimated is represented below:

Area	Formula	With Hybris DB tested
Hybris tablespace	DB_Size	145
System/UNDO table space	DB_Size * 0,5	73
REDO logs	DB_Size * 0,1	15
Archive Logs	DB_Size * 1.0	145
Backup	DB_Size * 2.0	300

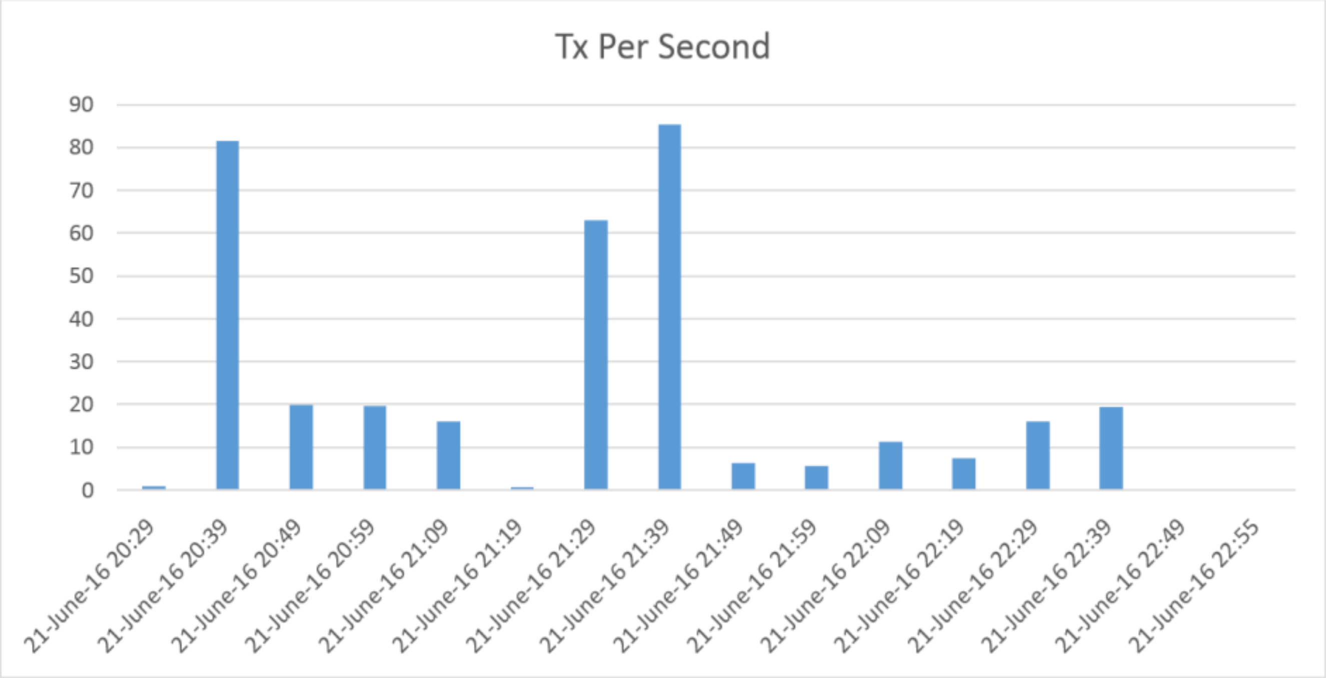
In total 700 MB should be reserved for Hybris when allocating disk space for these example implementations.

Storage access

From this we could estimate or measure TPS values for different usage profiles. With Oracle, in typical OLTP use, 30 IOPS per transaction is good number to consider.
For query performance, it is more complicated to say, because part of the data queries fetch are not hitting the disk at all. This is due to Oracle buffer cache that stores pages fetched from disk in memory. Only invalidated pages are accessed. Depending on buffer cache hit ration, queries

either hit cache (logical read) or access disk directly the disk (physical read). In some cases, like in case of full table scan, queries might bypass buffer cache and access the pages on disk directly and load them to PGA memory (direct path read for PGA memory area in user session).

Following diagram illustrates transactions per second generated by catalog import and sync.



For example, for catalog import and synchronization workload we can estimate the amount IOPS needed from disk. This workload does 50-100 transactions per second. If each transactions generated 30 IOPS on disk access, this estimation would give us 1500-3000 IOPS and this would be minimum requirement for storage system performance. As we can see from diagram above, transactions per second can peak up to 100 TPS. Therefore, to guarantee level of service we should provide at least 3000 (100*30) IOPS from storage to make sure that transactions do not have to queue for the disk.

CPU

Usually operating system gives one single number in percentage for CPU utilization. However, there are different types of CPU usage that together make this number:

- IO Wait, CPU is waiting for disk (or network) to come back with interrupt
- Host CPU, OS (kernel) is doing maintenance work and cleaning memory. For example page cache in Linux (pdflush).
- User CPU usage. This is the actual CPU used by Oracle process for querying and processing data.

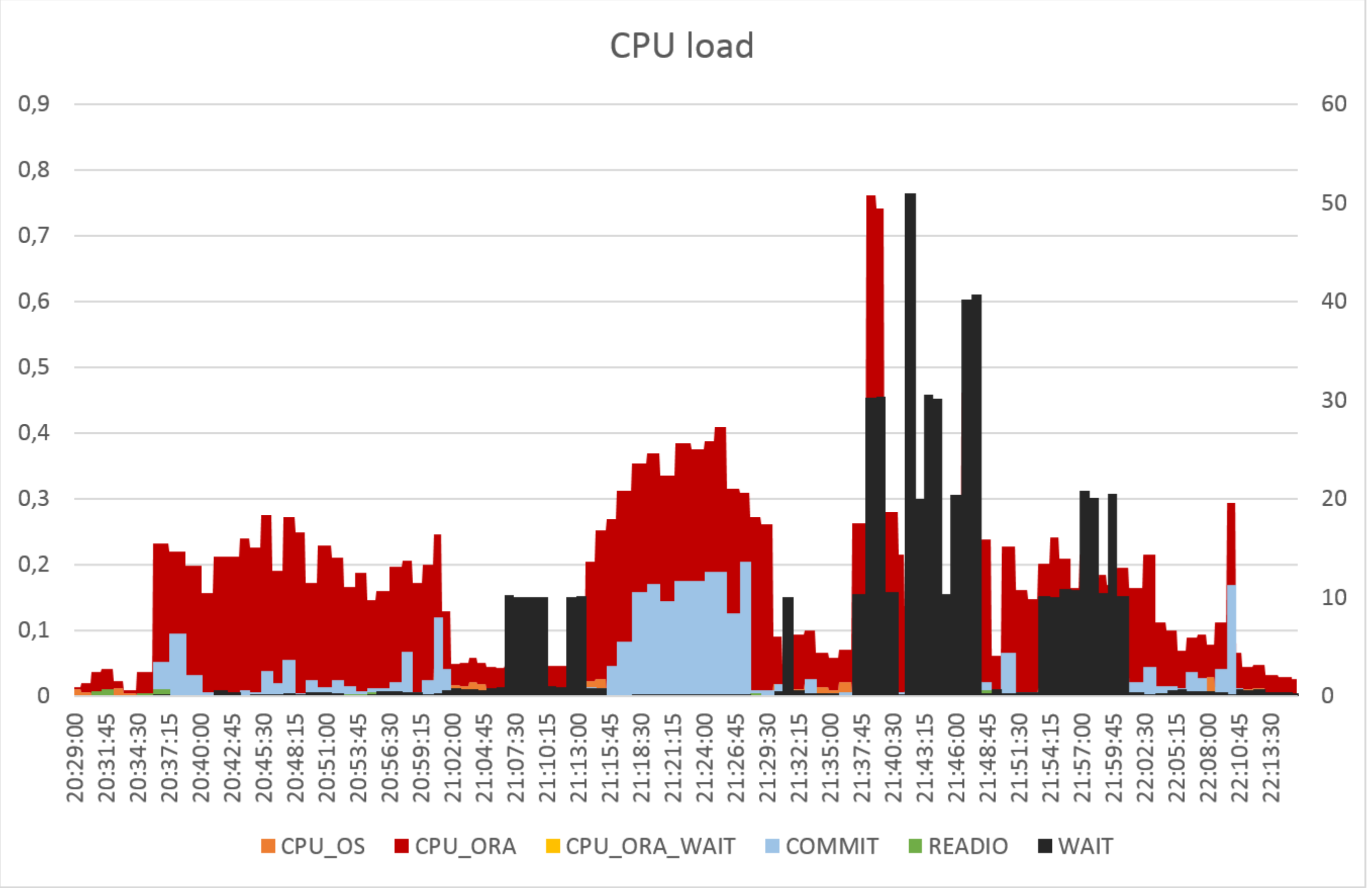


Diagram above is from data collected during Powertools catalog – import and synchronization. This test was done on virtualized system sharing i7 – processor with Hybris and database. What we see from the picture is that CPU in database is not pushed to limits during the import and catalog sync. This is probably due to network between Hybris and Oracle as Hybris cannot provide transactions on such frequency during catalog sync that it would push CPU usage higher.

From the diagram we can see that

- Oracle uses 30% of CPU available.
- During the run high amount of Commit – wait event is seen. This is when DB engine writes to REDO – log to disk. REDO log contains history of all changes done in DB.
- High amount of wait at the end of sync period. Database is likely flushing the changes from memory to disk.

What affects to CPU Usage:

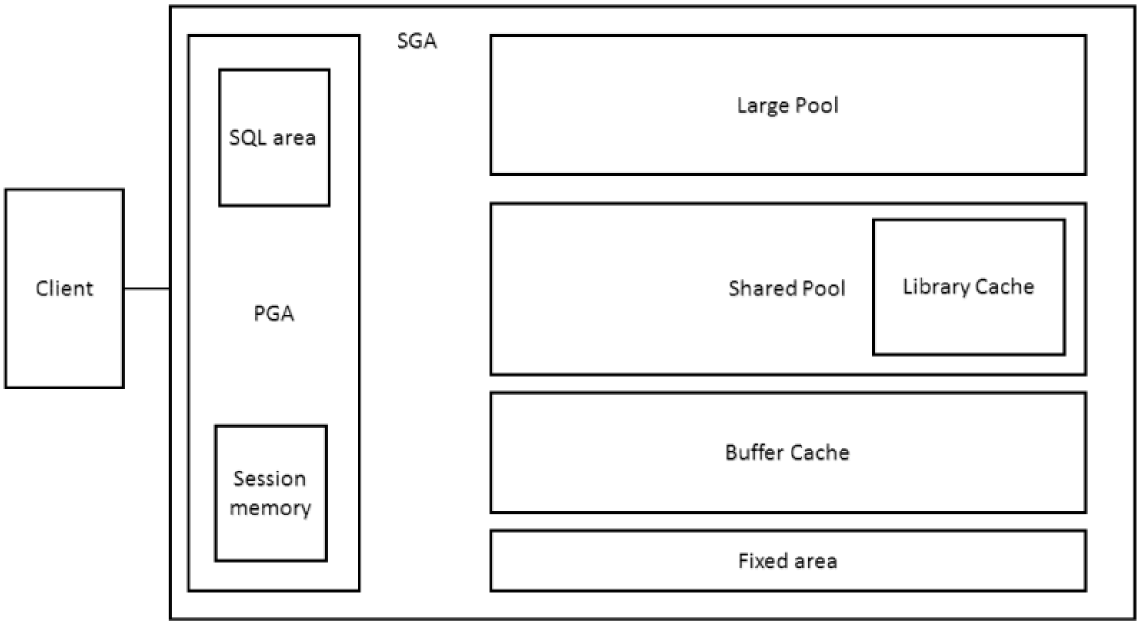
- Operations that have impact on Oracle process CPU utilization:
    - Large sorts
    - Parsing of queries
    - Mathematic functions in rows/columns
    - Index scanning
- Operations that affect to CPU WAIT
    - Accessing storage system; reading pages and flushing dirty pages from memory to disk
    - Network
    - Contention for CPU or other resources on database server
- CPU OS
    - Other processes

In this section, CPU utilization for only one workload scenario is explained. This is because database CPU usage for single user on browsing the storefront was analysed but it had minimal or no impact on CPU load due to efficient caching Hybris side. Therefore, correlation between Hybris storefront usage and Oracle database CPU utilization still remains to be envisaged later.

To estimate overall database CPU usage for Hybris, one would have to include multiple realistic workload scenarios and then sum up the CPU time spent within the desired time window (i.e, one hour). Wait events for storage should be minimized to understand real CPU usage.

Memory

Internal Oracle memory structure is defined in diagram given below.



Memory used by Oracle process is sum of PGA and SGA sizes.

PGA

PGA contains process memory for single connected user. In Hybris case, there is 90 connections typically in connection pool. This pool is theoretically capable of providing connections up to 90 PI/s and recommendation is not usually to enlarge the pool. Hybris storefront queries have minimal memory usage as the queries are simple, use query parameters instead of literals and hit indices. Typically, types are search with primary key and sorting is rare.
Therefore, rule of thumb to estimate PGA size is: for one connected user 1 – 2 MB of PGA memory to store most frequently used queries and results. In storefront use, it is likely that same SQL-statements are repeated on high frequency.
In this case, making sure that PGA has at least 500 MB is good starting point for performance tuning.

SGA

SGA holds shared areas in memory that all clients share. Most important of them is buffer cache that caches data fetched from the disk that contains indecies and tables. Depending on size of the database, at least most frequently and small sized tables should be pinned to buffer cache so that they are always available for in memory read. Other areas are, for example, shared SQL - statement pool where pre-processed and precompiled statements are stored to be shared with users (Library cache in picture above). Storing and sharing precompiled SQLs in library cache is why one needs to use query params instead of literals as then statements can be shared.

Sizing of the SGA area has direct impact of CPU (query parsing) and IOPS executed to disk storage from database as more data can be retrieved from cache. Sizing of SGA depends on amount of CPUs available. Larger the SGA is, more computing power is needed to maintain the SGA structure in memory.

Oracle has automatic memory management (AMM) functionality to configure sizes of different areas of the memory. Maximum memory assigned for Oracle, and targets for SGA and PGA can be set with configuration parameters. Increasing SGA target when enough CPU and memory is available may improve database performance and reduce physical disk reads done by the database. Oracle has views v$pgastat and v$sgastat to monitor current PGA and SGA sizes in memory.

References

Material used for this page:

1. "Sizing Oracle on Microsoft Windows and Dell PowerEdge Servers, a White Paper sponsored by Microsoft, Dell and Oracle", By Larry Pedigo, Performance Tuning Corporation, December 2004
2. "Dell EqualLogic Best Practices Series, Sizing and Best Practices for Deploying Oracle 11g Transaction Processing Databases on Dell EqualLogic Storage, Chidambara Shashikiran"
3. "Determining the IOPS Needs for Oracle Database on AWS", Abdul Sathar Sait and Jinyoung Jung , December 2014
4. "Running Applications on Oracle Exadata"
5. "Oracle Exadata Recipes: A ProblemSolution Approach"
6. http://www.moviri.com/2014/02/demystifying-oracle-database-capacity-management-with-workload-characterization/
7. http://www.slideshare.net/RenatoBonomini/ccmg-movirioracleworkloadcharacterization