solarwinds

ORACLE EXADATA PERFORMANCE

Yes. You still need to tune.

By Kathy Gibbs, Product Manager

Presentation to NYOUG Winter General Meeting 12/2011

INTRODUCTION

Exadata is a bit of a rock star in the Oracle world right now. Everyone wants to jump on board and get it installed in their environment. One of the selling points of Exadata is that you just install it, drop all your indexes and it will increase your database performance significantly all within about a week. Although part of this statement is correct, if you only follow these steps, not only do you open the door for specific queries to run slower at the start, but there is a good chance even some of the queries that are running ok now will exponentially get worse. The old adage is true: bad code and bad architecture, even on Exadata, is still bad.

EXADATA OVERVIEW

There is more than one 'Exa' product so we will take a moment to briefly discuss the other products Oracle is currently offering.

Exalogic with the Elastic Cloud is the machine for application servers. One way I distinguish Exalogic from the other products is to equate 'Weblogic' and 'Exalogic'.

Exalytics is the In-Memory machine which is configured for analytics. It contains an 'optimized' TimesTen database, BI Enterprise edition and Essbase. The rumor on the blogs about this machine is that Oracle is competing with SAP's HANA hardware. This is because Oracle originally described the Exadata Database Machine as a solution for data warehouses, and some people still wonder about the distinction.

Lastly, the Database Appliance which is different from the Database Machine which is what Exadata is often called, though lumped in with the other 'Exa' products, really isn't part of this family. The Database Appliance is really just a preconfigured fully redundant hardware database solution with 11gR2. It comes with RAC and RAC One. I included the Database Machine here because since it is a new hardware offering from Oracle, there is sometimes confusion.

WHAT IS EXADATA?

Exadata is a preconfigured combination of hardware and software that provides a platform for running the Oracle Database (only 11gR2 and above). Since Exadata Database Machine includes a storage subsystem, new software has been developed to run at the storage layer. This has allowed developers to do some things that are just not possible on other platforms.

The first Exadata platform was released in 2008. This was called Exadata V1 and was strictly on HP hardware. This first version was strictly configured for large data warehouse environments and was adopted mostly by large telecom shops. In 2009, Oracle released Exadata V2 with some significant changes. First the platform switched from HP to exclusively Sun. Secondly they added in

Smart Flash Cache technology which in a large part allowed this version to be marketed to not only large data warehouse shops, but any OLTP shop as well. In 2010 Oracle completed the acquisition of Sun and then came out with X2-2 and X2-8. These updates were primarily hardware focused.

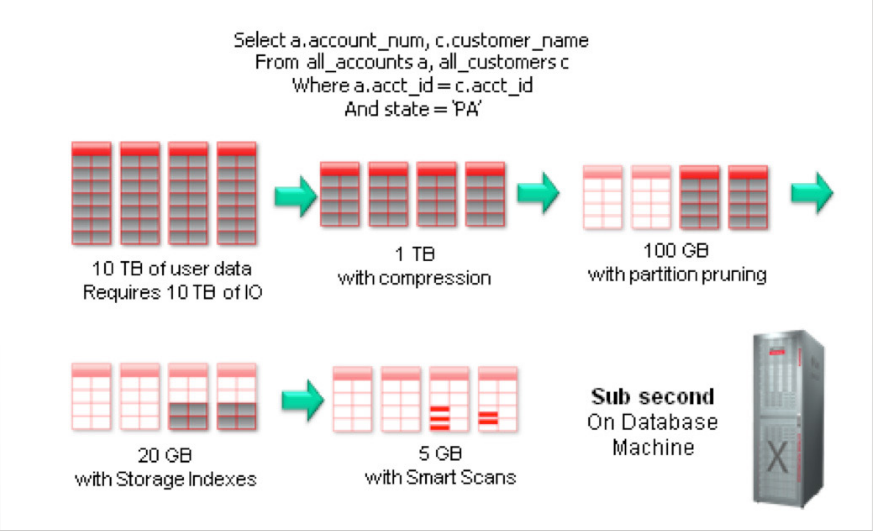
EXADATA PERFORMANCE

Exadata performance can be broken into a couple subsets.

- Cell Offloading
  - Smart Scan
  - Storage Indexes
  - Hybrid Columnar Compression
- DBRM/IORM
- Smart Flash Cache
- Additional views to monitor and query

Cell offloading or just offloading is the generic term to describe what is now being done at the storage tier that would have been done at the database tier before Exadata. This is probably the greatest advantage when buying Exadata, so it is something that people want to see reported on. It is considered an advantage not only because so much of the 'magic' of Exadata happens at the cell level but also because licensing at a cell level is much cheaper (8 to 1) than the database licenses.

For the Smart Scan, part of the Exadata machine operations returns less data to your database, decreasing I/O. You can monitor the Smart Scan by two main wait events which are 'Cell Smart Table Scan' and 'Cell Smart Index Scan'. It is good to note that the flow of data from Smart Scan can't be buffered in the SGA pool, which is part of the reasons this returns data faster.



```
Select a.account_num, c.customer_name
From all_accounts a, all_customers c
Where a.acct_id = c.acct_id
And state = 'PA'
```

Figure 1: *Why Exadata is so Efficient - Smart Scan*

Storage indexes go hand in hand with the Smart Scan. The goal for the storage indexes is not to reduce the amount of data; they are designed to eliminate time spent reading data. A storage index is an in-memory structure that holds some information about the data inside specified regions of the physical cell storage. Think of these as a pre-filter since they pass the query predicates to the storage servers and storage indexes contain a map of values in each 1 MB storage region. Any region that can't possibly contain a matching row can be eliminated without ever being read. You can also think of the storage indexes as a type of partitioning. Disk I/O is eliminated in an analogous fashion to partition elimination. If a partition can't contain any records of interest, the partition's blocks will not be read. Similarly if a storage region can't contain any records of interest that storage region need not be read. They are often used with OLAP systems. You can't create a Storage Index per se; it is automatically created after the storage cells receive repetitive requests. Because these indexes are stored in memory, the indexes are gone and have to be rebuilt after a reboot of Exadata.

Hybrid Columnar Compression or HCC is a technology that utilizes a combination of both row and columnar methods for storing data. It does this by creating a logical construct called a 'compression unit' which stores a set of HCC rows. When data is loaded, column values for a set of rows are grouped together and compressed. After the column data for a set of rows has been compressed, it is stored in a compression unit. To maximize speed of loading, you should use 'DW' loading techniques (i.e. Insert statements with APPEND hint, parallel DML, direct path sqlldr, CTAS.) Although you can use conventional DML (Data Manipulation Language) operations, they just may load slower. There are three types of HCC, the first Query Low uses the LZ0 compression algorithm and it is expected to have a 4x compression ratio. The second is Query High which uses zlib or gzip to get a 6x compression ratio. The third is Archive Low which also uses gzip but at a higher compression level. This third option is harder to load so it is not used unless you are going after partitions or data sets that aren't being updated.

DBRM/IORM

One of the issues that will complicate an Exadata initiative is moving from single boxes for applications to a consolidated hardware platform. A user has a false impression that if they used to have a host for its application alone it will run slower on a consolidated hardware solution. Database Resource Manager (DBRM) and the I/O Resource Monitor (IORM) is the way to solve this performance concern. Without using resource manager, all of sessions are given equal priority, and one high activity database can monopolize resources so it is a very critical step in your Exadata implementation.

DBRM is the same resource manager you are currently familiar with. One thing new with 11gr2 is instance caging which provisions CPU usage at a database level. As stated, this allows you to cage or control CPU use for a single instance of a database. To see if this is causing a problem, you'll want to check for the wait event 'Resmgr: cpu quantum'. If this event is high it means DBRM is actively throttling CPU consumption. In order to ensure the resource plans are being used to how you want, they are some V$RSRC views to monitor.

IORM is strictly for Exadata and there are a couple of items to discuss here to help isolate database load. Interdatabase IORM manages priority among multiple databases by database name in combination with work priority. The second item is IORM categories, this is a new attribute, but still utilized by isolating on database name. As an example you could create a category called 'oltp_category_batch' that is for two CPUs. You would then be able to assign one-to-many databases to it. The third option or setting is for Intradabase IORM. On Exadata when a DBRM plan is activated the db transmits a description of this plan to all cells in the storage grid. So in a way this is a bit of the 'default' as long as a DBRM has been created first. No matter what option you use, IORM is managed at the storage cell level. For each cell disk cellsrv maintains an IORM queue for each consumer group and each background process for each database accessing the cell.

```
Cellcli> alter iormplan objective = low_latency
Cellcli> list iormplan attributes object
low_latency
```

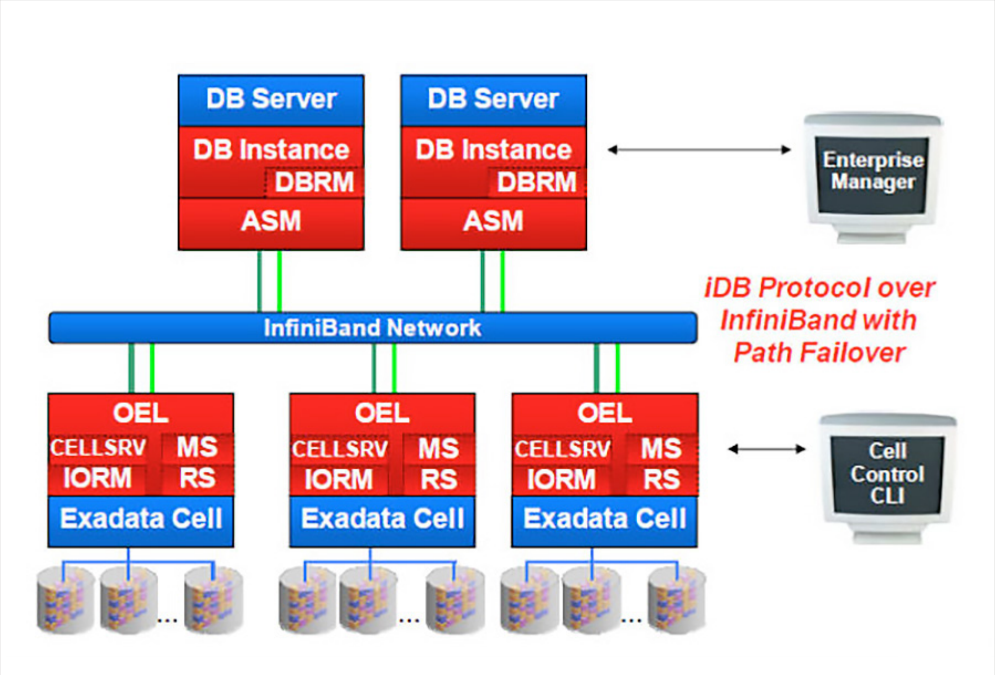


Figure 2: *Exadata Software Architecture*

SMART FLASH CACHE

Smart Flash Cache is different from Flash Cache which is available for all 11gr2 databases. On a full rack of Exadata, Smart Flash Cache offers an additional 5TB and it can be used in two ways. The first is to be configured as cache which is really the more recommended configuration. The second way to use Smart Flash Cache is to carve it out as a solid state disk that ASM can use. This is not really the recommended use for this storage, but there are instructions on how to set it up[1]. Most the time I have seen this used for redo or temporary tablespaces, but every time I have read something about it, the performance gains have been minimal versus using the Smart Flash Cache as additional cache. In order to use this, you do have to set up the init parameter cell_flash_cache=keep. Cellcli can be used to monitor the flash cache usage as well as v$sysstat.

Cellcli CELL_FLASH_CACHE=KEEP

- Create flashcache all size=300g
- List flashcache detail
- List celldisk attributes name, diskType, size where name like 'FD.*'
- List metriccurent where objecttype = 'FLASHCACHE'
- List flashcachecontent – shows objects in cache

WHAT TO LOOK FOR WHEN TUNING

The steps to follow when you are going to tune statements on Exadata are first to look at response time and to monitor that. You should first focus on your high response time queries and work your way from highest wait time to lowest. The next item to look at is the database layer, focusing on utilization and efficiency. The next level to monitor is the storage cell layer for utilization and efficiency. Once that has been tuned or there are no major issues you can start in on the advanced metrics and monitoring specifically for Exadata. Through many of these options you will look at v$views, CELLCLI, and OSWatcher. There are specific wait events to look for when on Exadata. It is important to note that there are no additional wait events built into the database code for Exadata. All the waits events are in all the database code, it is just that no time is allocated on other systems that are Exadata specific. There are too many to list here, if you run the query below you will find most of the events:

```
Select event operation, count(*) from (
  select sql_id, event, sql_plan_operation||''||sql_plan_options operation
      from DBA_HIST_ACTIVE_SESS_HISTORY
      where event like 'cell %'
      group by operation, event
      order by 1,2,3
```

EXADATA CAUSING PERFORMANCE ISSUES

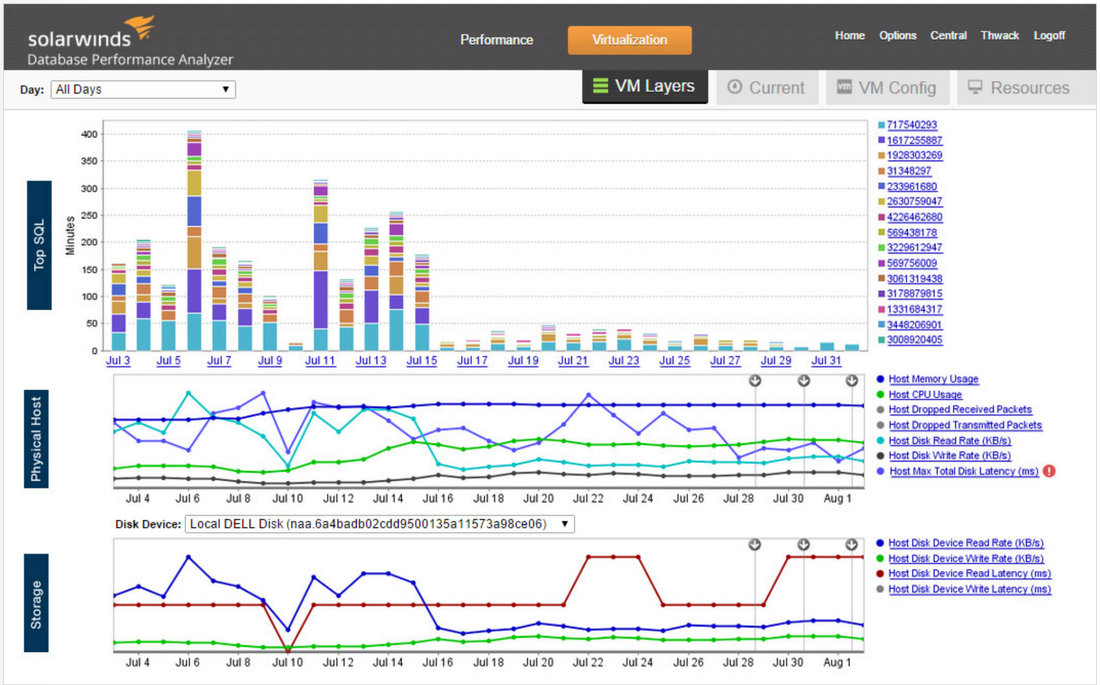
No, this is not any reason not to get Exadata, this section is just to understand that there are times when Exadata can actually cause performance problems. Oracle states to drop all indexes when you move to Exadata. In an OLAP system there are many situations where this actually works and speeds up your query but there are some situations if you blindly drop indexes performance will take a hit. If you have a finely tuned query especially on OLTP you may not see any benefits on Exadata. These are the exceptions, but everyone I have talked to has had that one query that is a pain point.

CONCLUSION

Simply put, Exadata houses an Oracle database. Remember that when things get overwhelming. It is just another database. When you are tuning, always start with cell offloading. You will see the biggest benefit and often that is what executives like to see working due to licensing constraints. Besides learning and understand roles around Exadata, in my personal case the biggest problem came from upgrading our existing database from 10.2.0.4 to 11gr2. Remember the DBRM, IORM and flash cache. They may take a little longer to set up, but they are worth it in the end. Always be aware that Exadata is not the "be all, end all"—you still need to tune.

HOW CAN DATABASE PERFORMANCE ANALYZER HELP?

Database Performance Analyzer (DPA) from SolarWinds (NYSE: SWI) provides the fastest way to identify and resolve database performance issues. DPA is part of the SolarWinds family of powerful and affordable IT solutions that eliminate the complexity in IT management software. DPA's unique Multi-dimensional Database Performance Analysis enables you to quickly get to the root of database problems that impact application performance with continuous monitoring of SQL Server, Oracle, SAP ASE and DB2 databases on physical, Cloud-based and VMware servers.



Day: All Days ▾

Disk Device: Local DELL Disk (naa.6a4badb02cdd9500135a11573a98ce06) ▾

For additional information, please contact SolarWinds at 866.530.8100 or e-mail **sales@solarwinds.com.**

[1] http://www.oracle.com/technetwork/articles/systems-hardware-architecture/oracle-db-smart-flash-cache-175588.pdf