

Introduction

Exadata Machine was first introduced by Oracle in 2008 and now it has become one of the most popular database platform to host Oracle databases. Exadata machine is like a mini data center in itself, comprising of Database Servers, Storage Servers, InfiniBand Switches and Ethernet switch working together to deliver outstanding database performance.

A number of Oracle customers are currently trying to implement Oracle Exadata on their own. There can be number of reasons for this. Most of the times, they insist in working on their own, simply because they are unwilling to spend a substantial amount of money on hiring an Oracle Exadata expert. I don't understand the need to save money, since you are already paying a handsome amount to buy Exadata Machine and have experience team of DBA's working for your organization. But if you don't follow a proper implementation strategy, you most likely to end up doing just a database migration. It will be a mistake to treat Exadata machine like any other hardware, otherwise you will short of fully utilizing Exadata machine great feature like storage indexes, smart scan and offloading. Such kind of Oracle Exadata implementation (more like Exadata Migration), can also be extremely costly and will not provide great ROI (Return on Investment) as well. Eventually, customer end up have over utilizing some of very costly hardware resources and buying additional Exadata Hardware and software they don't need.

Implementation Strategy Overview

Purpose of this article is to provide readers Exadata implementation strategy which can lead to successful Exadata project. Purposed strategy is made up of 5 implementation phases Planning, Migrating, Optimization, Testing and the Final Cutover respectively.

Implementation Strategy



Plan



Migrate



Optimization



Test



Cutover

Basically each Exadata implementation should start with planning. Then move to migrating data using suitable migration methods. Once data migration is completed, look into implementing Exadata best practices and features to achieve extreme performance from your Exadata machine. Each migration should also go through multiple testing cycles before the cutover.

Phase # 1: Plan

During planning phase, current system which is targeted for migration should be analyzed in details. Collect information like database size, application type, I/O throughput and memory footprint. Then based on your analysis make important deployment decisions like, Do you need to virtualize Exadata Machine? ,do you need to implement resource management? What ASM Redundancy level should you choose? And how big should be your FRA? Migration strategy should be discussed in early phases of Exadata implementation. Map out each database migration with a particular migration method like Golden Gate, Export Import or data Guard. You should also inform business users and other stake holders about upcoming migration, so they can plan for outage. High availability options should also be discussed early based on SLA requirements. Oracle Exadata does comes with Oracle Real application cluster high availability option but if you migrating from a non-cluster, make sure your application is design to handle this architectural change. Testing is also an important part of Exadata Implementation, make sure to discuss testing options with stakeholders and come up details testing plans.

Plan



Phase # 2: Migrate

There are several ways to migrate databases to Exadata Machine. Data migration methods can be categorized as physical migration or logical migration. Each migration method has its own pros and cons, so analyze them carefully based on your requirements.

Here are two of the most widely used Exadata migration methods:

1. **Logical Migration**
2. **Physical Migration**

Migrate



Logical Migration

Data migration using Data Pump, Golden Gate, and logical standby are considered logical database migration. This migration method is particularly useful, if you want to create Exadata database using DBCA Template with all the Exadata best practices built in it. Additionally this method is useful when you are upgrading database version during migration and Big Endian to Little Endian format. Here are brief description of some logical migration methods available to you with their benefits.

Published in IOUG SELECT Magazine

Logical Standby: Logical standby databases is an option when physical structure of the source database don't have to match with target database. The Oracle Logical Standby migration technique is generally best known for the following benefits: Min Downtime, Adjusting ASM AU size, Implement physical changes to database during migration.

Golden Gate: This strategy is known for its transition flexibility. Database from any platform, any endian format or any database version can be effortlessly migrated to Exadata with minimum down time. Benefits include: Min Downtime, cross platform migration, a zero data loss and fallback plan

Data Pump: The Data Pump approach is the most common and widely used strategy for migrating Oracle databases. With data pump, you can migrate almost any Oracle version, from any platform to Exadata. This is also a preferred method, if you want to compress or partition your tables during migration. It provides you the migration benefits of: Simple migration, Full Data Type Support and Cross Platform Support.

Physical Migration

Physical migration usually mean block by block copy of source database running on Exadata machine. Like logical migration, it has its own pro and cons and in some cases this methods can very useful. This is migration methods is suitable for customers who are not planning to introduce any new database features like Compression or partitioning. Even than you can still introduce new database features after the migration. Although, there is one major concern about using physical migration method, since it is a block by block copy, you will be bringing in all the characteristics of a source database. Exadata comes with its own set of best practices and your source database might not in line with them. I strongly suggest running Exacheck utility after physical migration which is design to evaluate hardware & software configuration, MAA Best practices and database critical issues for all Oracle Engineered Systems. Here are brief description of physical migration methods available to you with their benefits.

Physical Standby: The physical standby database typically requires very little downtime, you simply needs you to create and configure the Data Guard to the target Exadata database machine. When ready, you simply have to perform the switchover for complete the migration. This strategy works well for a same release, with supported cross platforms. It provides you the migration benefits Min Downtime and simplicity.

Transportable Database (TTE): The Transposable database is the best type of strategy for migrating towards a different platform, with the same endian format. It provides you the migration benefits of simplicity.

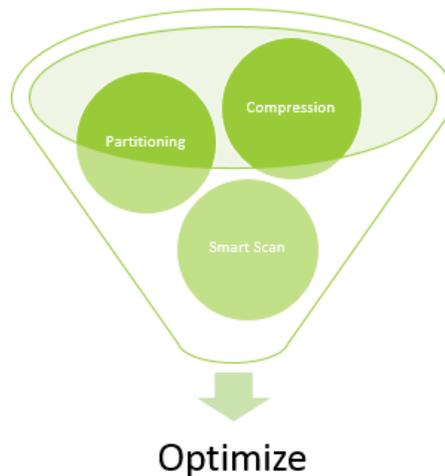
Published in IOUG SELECT Magazine

Transportable Tablespace (TDB): Works best when you want to migrate to a different platform, with a different endian format and a different release. It provides you the migration benefits of simplicity and support for cross platform.

Phase # 3: Optimize

If you really want to take full advantage to Exadata Machine, you should look into implementing some of the following database / Exadata Features. For Example: Compression not only reduce your storage footprint but also improve performance. Partitioning will also improve performance, provide you maintenance advantage and increased availability. Parallel execution will help you with performance, same goes for properly caching tables to Exadata Flash Cache. Even though offloading and smart scan are enable by default but make sure they are happening. Exadata Machine does a great job managing resources by itself but if planning to use Exadata as consolidation platform, you should look into implementing resource management through DBRM & IORM. Here is the brief description of some Exadata and database features and how they can play a key role in achieve extreme performance from your Exadata Machine.

Optimize



Published in IOUG SELECT Magazine

Compression: Regardless of Exadata, Oracle has two native compression types, basic table compression and OLTP Compression. You will not get good compression ratio with Basic table compression and it will not support DML operations. But you can get reasonable compression ratio with OLTP compression ratio and it will also support DML operations. Exadata comes with its own compression called hybrid columnar compression. You can get extremely good compression ratio with Hybrid Columnar compression but OLTP operations are not supported. Compression not only save some very expensive Exadata storage, it also improve performance by efficiently utilizing database buffer cache.

Partitioning: Oracle support many types of partitioning techniques including range, list and hash. Partitioning can help you achieve better performance through partition pruning. You can perform certain maintenance tasks like truncate a partition, gathering stats and rebuilding a local index just for a particular partition, hence providing you ease of maintenance and high availability.

Parallelism: You can execute your queries in parallel to speed up your load. If you are not already using parallel query feature, you should look into introducing this feature during or after the migration. You can enable parallel query execution at object level or you can use SQL hint. You can also let Oracle determine the degree of parallelism based on a set of criteria and some initialization parameter settings. This feature is called AUTO DOP and it will automatically parallelize your queries based on a threshold. The threshold that is prominently mentioned above is set by `parallel_min_time_threshold`. The default of this parameter is 10 seconds. If you want to run more statements in parallel, make sure to reduce that number so that more plans can qualify for parallel evaluation.

Flash Cache: Exadata comes with tera bytes of flash cache. It also called smart flash cache because it has an ability to move data in and out from cache based on usage. Its enable by default, you don't have to configure anything to enable it but you can turn it off or encourage caching a particular object by alter table statement. Flash cache also comes with Write back option, which provides the ability to write I/O directly to PCI flash in addition to read I/O. If you have write intensive application and one finds significant "free buffer waits" or high IO response times, then write back flash cache is a suitable option.

Offloading/Smart Scan: Exadata extreme performance is archive though offloading and smart scan. Offloading mean some of Oracle process are offloaded to Exadata Storage nodes. Oracle processes that can be offloaded to storage nodes are incremental backups, datafile creation, decompression and decryption. Oracle smart scan refer to an Exadata capability of performing projections and predicate filtering operations , when mean storage layer will only return required rows and column to database nodes , hence reducing I/O and network traffic between storage server and database nodes. There are some pre-requisites for smart scan like direct path read and full table scan, so make sure smart scan is happen for your database and you are able to offload decryption and decompression to storage nodes.

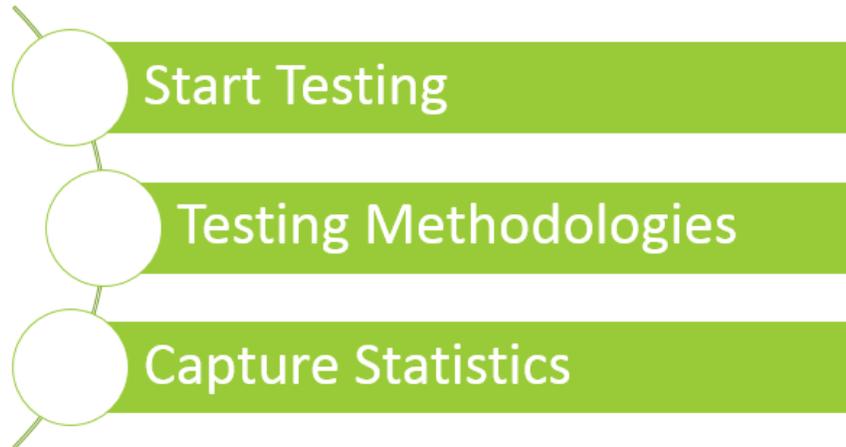
Published in IOUG SELECT Magazine

Resource Management: If you are planning to consolidate databases to Exadata platform, you might want to look into implement some level of resource management, so you get consistent performance across different workload and databases. You can use Oracle native resource management tool called DBRM to manage CPU utilization, parallel queueing and long running queries and you can use IORM Exadata native utility to manage I/O throughput and latency.

Phase # 4: Testing

Testing, Testing and Testing. Testing is one of the most important part of Exadata implementation strategy. You should have test plan ready before you started the migration process. There are many types of testing you can perform during this phase like performance test, break test and failover test and you should at least test all of your critical processes. If you are planning to introduce new features like compression or partitioning, make sure to alter your test plans to compensate for these changes. Even though these features are design to improve database performance but sometime these changes can cause SQL queries to behave badly. Make sure to capture performance stats using tools like AWR, ASH, and SQL Performance analyzer and compare it with your baseline. AWR reports will provide you all the details you need to compare elapse time, IO wait and CPU utilization. Also compare critical processes and queries using ASH reports, it will provide you further details about execution plans and wait times. Also validate Exadata configuration through running Exachk and remediate any issues you encountered during this phase.

Test



Phase # 5: Cutover

The final implementation phase is the cutover. Make sure to backup both source and target databases and have a fallback plan, just in case if you encounter any issues after the cutover. If you are migrating a customer facing critical database, fallback plan should include syncing data back to source database using replication technologies. Depending on migration method and maintenance window, you will probably have to sync your target database just before the cutover. You will need to perform data validation, especially if you have used any other methods, than physical standby, to sync source and target databases. You should be on guard for next 48 hours and ready to remediate any issues.

Cutover

Source



Exadata



Conclusion

Described above was the tested Oracle Exadata implementation strategy. If you follow above techniques and strategy step-by-step, you will undoubtedly be able implement Exadata with all advantages. Without a proper implementation strategy, you will be just performing a data migration and most likely to miss out opportunities to implement extraordinary Exadata Machine features. Good Luck.