An Oracle White Paper
March 2015

A Real-World Technical Case Study of a Global Manufacturer

Oracle E-Business Suite, Oracle Exadata, and the Oracle Maximum Availability Architecture

Summary

One of the world's largest implementations of E-Business Suite was successfully upgraded and migrated from a legacy environment to Oracle Exadata Database Machine and Sun application servers. The customer's new environment makes extensive use of Oracle Maximum Availability Architecture to provide high availability, high performance, and comprehensive data protection in a consolidated environment characterized by a high volume of mixed workloads (OLTP, reporting, and batch) and demanding service level expectations. To maximize the use of their computing assets, the customer implemented a very cost-effective architecture by creatively using Data Guard standby systems for multiple purposes and incorporating low-cost Sun ZFS storage.

Benefits of the new architecture include:

- Despite almost 5 times increase in IO load in the upgraded applications, the IO response time improved by a factor of five on the Exadata systems.[1]
- Transaction commit response times are over 5 times faster.[2]
- Exadata IO Resource Management enabled a highly mixed workload that provided excellent response times for interactive OLTP while simultaneously running batch, high speed reports, and database backup.
- Data Guard Physical Standby provides a maximum data loss exposure of 30 seconds regardless of the nature or scale of outage (from a disk failure to a complete site outage).

[1] Lower average wait comparing 2.5ms on HP PA-RISC to maximum average wait of .5ms on Exadata during first quarter close.

[2] Lower average wait comparing 4.87ms on HP PA-RISC to maximum average wait of .874ms on Exadata during first quarter close.

Intended Audience

Readers of this paper are assumed to have experience with Oracle Database 11g technologies, familiarity with the Oracle Maximum Availability Architecture (MAA), and a general technical understanding of Oracle Exadata Database Machine. In-depth background on these topics is covered in other documentation and technical white papers available on the Oracle Technology Network[3]. This paper will provide configuration details and benefits specific to the production deployment being discussed. This is a real example of customer experience, an internationally recognized global manufacturer, deploying E-Business Suite on Exadata and following MAA best practices. Please see the Appendix for a list of recommended technology white papers and acronyms used in this paper.

[3] http://www.oracle.com/technetwork/database/exadata/index.html

Introduction

The customer was reaching end of life support for their hardware platform as well as for their primary E-Business Suite implementation. Virtually everything needed to be upgraded:

COMPONENT	ORIGINAL VERSION	FINAL VERSION
DB Server	3 HP PA RISCSuperdomes, EMC Symmetrix	Exadata X2-8
Database	10.2.0.3 with Real Application Clusters	11.2.0.3 with Real Application Clusters
Middle Tier Servers	HP RP8400	Sun X4170
Network	1GE	Infiniband and 10GE
Application	11.5.10.2	12.1.3

TABLE 1 – SCOPE OF UPGRADE

These basic requirements drove the project:

- Upgrade everything – hardware, operating system, database, middle tiers, network, application, customizations – at one time, to save the cost of testing a string of smaller but still significant environment changes;
- Increase capacity to handle new E-Business Suite R12.1.3 functionality that includes Sub-Ledger Accounting
- Provide excellent performance for the mission-critical manufacturing floors and back-office processing while also supporting a monthly/quarterly “8-Hour Close,” where all the processing to close financial books globally is done within one working day;
- Build a resilient architecture to assure availability in the face of both physical and logical failures;
- Optimize footprint, deployment, and ongoing management costs.

The customer successfully deployed Oracle Exadata Database Machines for their primary E-Business Suite implementation, hosting 115 E-Business Suite applications from Order

Management through Manufacturing to Financials – in essence a consolidated solution by itself – as well as the matching Oracle Advanced Planning application in a separate database on the same servers. The solution provides the capacity to support their daily operations even during their hallmark 8-hour close.

The Project

To provide the availability required by such a mission critical system, the customer implemented an architecture that included Oracle Real Application Clusters (RAC) as well as two Data Guard physical standby databases – one local to the production data center for recovery from operations-induced data corruptions, and one remote for disaster recovery (DR). To help manage costs, the customer consolidated QA/performance test, integration test, and development environments onto the same environment used for disaster recovery.

The new implementation is running E-Business Suite R12.1.3, an upgrade from E-Business Suite 11.5.10.2. Design changes in R12.1.3 result in a requirement for more computing resources. Exadata is easily handling the differences in peak workload:

	Exadata Total/Sec	HP PA-RISC Superdome Total/Sec
Redo size	4,672,847.40	4,461,880.62
Logical reads	4,374,800.20	1,605,704.97
Block changes	27,125.60	28,197.39
Physical reads	148,429.80	31,682.88
Physical writes	7,015.50	1,410.93
User calls	8,237.40	5,124.53
Parses	3,784.80	2,148.85
Hard parses	74.30	47.91
Logons	180.50	8.81
Executes	103,066.50	11,149.43
Transactions	253.30	133.97

TABLE 2 – LOAD PROFILE COMPARISON, HP TO EXADATA DATABASE MACHINE

The Exadata Machine cuts I/O wait times to 20% of those experienced on HP and increases productive DB CPU time by 42% while running the new heavier workload.

Along with the improved batch performance, online users enjoy consistent – and consistently good – response times.

This paper describes the customer's implementation at a high level and provides in-depth insight to the technical solutions used to implement key features in their new architecture:

- Safely using Exadata's Smart Scan feature and taking backups on the Exadata Machines while simultaneously serving the mission-critical OLTP load that keeps the customer's business running.
- Safely sharing a single Exadata machine for disaster recovery and development, test, and quality assurance / performance test databases for maximum return on investment in the disaster recovery system.
- Using Data Guard Physical Standby to provide protection against disasters, data corruptions, and operations errors that would otherwise lead to unacceptable downtime and data loss. The recovery point objective for these very busy systems allows for a maximum data loss exposure of 30 seconds regardless of the nature or scale of outage (from a disk failure to a complete site outage).

Architecture

The new environment supports the customer's largest manufacturing plants as well as their corporate operations. The challenges to be met with the implementation:

- Be fast: Provide compute support for the customer's mission-critical manufacturing floors as well as all back-office processing. While supporting daily operations, process the customer's hallmark "8-hour close." Do all this without negatively impacting user response times.
- Be available: Provide redundancy to protect against hardware failures, provide for disaster recovery, provide for reduced planned maintenance outage times, provide for recovery from logical data corruptions.
- Be cost-effective: Combine functions to maintain a rational footprint.

Primary Site - Production Environment

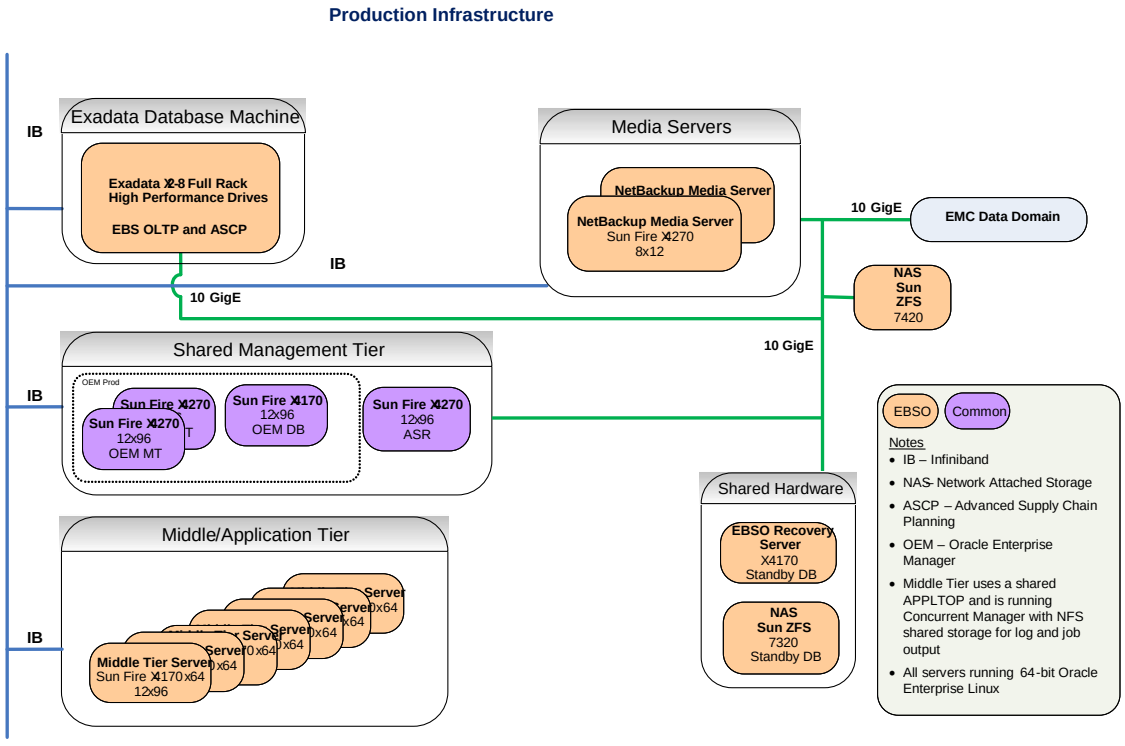

FIGURE 1 – EXADATA DATABASE MACHINE PRODUCTION ARCHITECTURE

The customer hosts their production E-Business Suite OLTP and Advanced Planning databases in a data center in the Northeast United States, on an Exadata X2-8.

Middle tier services, including the E-Business Suite itself, Oracle Enterprise Manager (OEM), Automated Service Request (ASR), media servers for backups, and the server managing a

local physical standby are all hosted on Sun X4170s or X4270s connected via Infiniband (IB) to Exadata.

For shared file system storage, there is a Sun ZFS 7420 for E-Business Suite and middle tier software, FTP services, and E-Business Suite Concurrent Manager Log and Out directories. The Sun ZFS 7420 is connected via IB to the middle tier fabric as well as via 10GigE to the backup server.

Weekly full and daily incremental backups are taken from the production database. The customer is using Symantec's NetBackup software to manage communications between Recovery Manager (RMAN) on Exadata and their Data Domain tape storage system. A pair of media servers was configured for redundancy.

The customer maintains a local Data Guard physical standby replica of their production E-Business Suite database as a data source to repair data corruptions resulting from operations errors. It is configured with a 6-hour apply delay (6 hours behind production), to provide quick access to older data. This database is managed from a Sun X4170, with the data stored on a Sun ZFS 7320, connected via a 10GigE network.

Disaster Recovery Site – DR and Non-Production Environment

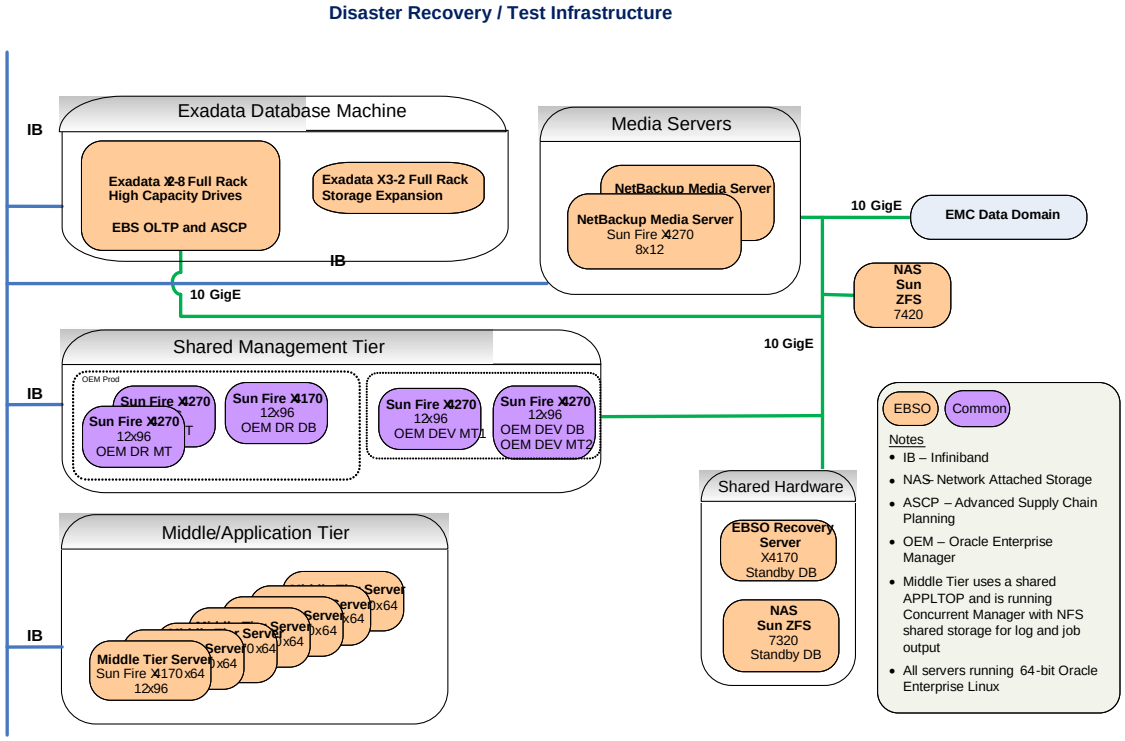

FIGURE 2 – EXADATA DATABASE MACHINE DISASTER RECOVERY/DEV/TEST ARCHITECTURE

The customer's DR environment is in a data center in the Southeast United States, over 600 miles from their production data center. It hosts databases that are Data Guard physical standbys of the production North America E-Business Suite OLTP and Advanced Planning databases. These standby databases use Data Guard real-time apply to be completely up-to-date and ready for immediate failover should the production environment become unusable. They are hosted on an Exadata X2-8 that is identical to production with the exception that high capacity (2 TB) disks were used instead of 600 GB high performance disks.

As in the production environment, Sun X4170s are used in the middle tier, with a Sun ZFS 7420 providing shared storage, all connected using InfiniBand. There is a Sun ZFS 7320 in the DR data center so the production configuration can be duplicated for test purposes, including the local standby.

Be Fast

Exadata has exceeded the customer's performance expectations. Exadata does this using:

- Exadata Smart Flash Log for low-latency commits,
- Exadata Smart Flash Cache for low latency reads,
- Smart Scans to offload and parallelize table scans and reduce network traffic,
- the high bandwidth / low latency IB network,
- Automatic Storage Management (ASM) that simplifies storage management while delivering the performance of a raw device, and
- Real Application Clusters (RAC) that scale performance while providing high availability.

The customer load-balances its entire online workload across both RAC nodes. The Advanced Planning database is configured to run on one node as required by the application, and the administrators have used the E-Business Suite concurrent manager node affinity feature to pin the job that integrates the two databases to the same node. Other than that, all concurrent manager jobs are load-balanced across both RAC nodes.

The customer has also implemented MAA best practices, in particular configuring huge pages for the shared global areas (SGAs), making the production SGAs very large, and melding the E-Business Suite and Exadata / MAA init.ora parameter recommendations.

This has resulted in consistently seeing "DB CPU" as the top event in the "Top 5 Wait Events" sections of the Application Workload Repository (AWR) reports – which means the Database Machine's CPUs are *not* waiting, they are working.

The customer's workload is largely OLTP based, with well-tuned key lookups. However, there are batch programs, Discoverer reports, and the occasional tuning problem that result in requests for scanning entire large tables – sometimes in a loop. Exadata's Smart Scan feature

returns this data quickly. See the next section of this paper for how to do this safely in an E-Business Suite environment – Managing Large and Small IOs on Exadata.

Be Available

The Exadata Database Machine implements redundant components so that services can continue to be available even if any given component fails. The customer uses MAA capabilities to fully utilize this redundancy and make component failures transparent to users, including RAC for its clustering services for the databases and ASM to mirror the disks.

The customer implemented high (triple) redundancy on all disk groups to protect against the possibility of multiple disk failures and also enable Exadata cell maintenance (e.g. battery replacement, software and firmware upgrades, complete cell replacement) to be performed in rolling fashion without compromising availability. Applications remain fully available even if a storage failure occurs while a cell is offline for maintenance.

They have also basically replicated their production install in another data center, where it can serve as a test environment for infrastructure changes and an environment for performance testing major new releases, as well as being an infrastructure to run production services if disaster strikes their primary data center. In this way their Data Guard standby databases reduce risks inherent in planned maintenance at the same time they provide DR.

They have implemented two Data Guard physical standby replicas of their OLTP production database. One is hosted in the production data center (the “local standby”) and is connected via 10GigE to the production Exadata Database Machine. While Data Guard transports redo directly from the primary’s memory to both standby databases, the local standby is configured with an apply delay to keep the data files 6 hours behind the primary (redo is written to Standby Redo Logs (SRLs) and archived to be applied after the rolling 6-hour delay has expired. If their production database suffers a logical corruption – e.g., someone drops or otherwise mangles data in a table – this database can be recovered to the appropriate point in time immediately prior to the corruption and data can be retrieved for the repair.

The disaster recovery standby is on the Exadata system in a data center located in the Southeast United States, over 600 miles away, and is always at ready to assume the primary production role if the primary site fails. The DR standby is kept current, receiving redo directly into its SRLs and applying the redo as it is received using Data Guard real-time apply. Tips about successfully configuring and managing multiple workloads in the standby environment (e.g. standby and test databases) are provided in a later section of this paper – Consolidating DR with Dev/Test/QA and Performance.

A brief description of the implementation of the two standby databases can be found in the section Data Guard and Network Configuration. The customer has documented and tested procedures for using each of these standby environments, covering complete failover of all production services to the Disaster Recovery data center, including all 3rd party applications.

Be Cost-Effective

While it is required to ensure adequate capacity to handle the required production load, it's also important to be pragmatic about the deployment. The customer clearly defined the requirements for each standby database (local and remote) and utilized a deployment model that achieved those requirements in the most cost effective manner. Their requirements influenced the chosen deployment model as follows:

- The local standby database will not be accessed via E-Business Suite software and does not need to serve a large number of users. It will only be opened for query when recovery is stopped, as the administrators need to freeze the image at a time before a corruption occurred in production. Thus the hardware assigned the task does not need to provide production capacity equivalent to the Exadata primary system. The database instance is being hosted on a Sun X4170, and the database itself resides on a Sun ZFS 7320.
- The remote standby database at the DR site is intended to run production if the primary site becomes unavailable – thus it needs to be configured with the same capacity of the primary Exadata system. Under normal conditions while production runs at the primary site, however, the standby only needs to support the relatively lightweight Data Guard managed recovery process. The excess CPU and memory capacity while the DR system is in standby role is very attractive to use for development and test environments. The ability to support additional databases, though, increases the storage capacity that is required relative to the primary. The customer determined they could configure the standby with high capacity disks with the addition of a full Exadata Storage Expansion Rack to provide both 1) enough disk space for all the test database copies (including standby databases for two of the test systems) and 2) enough IOPS capacity to run production without performance impact in the event of a disaster. During a disaster, the customer will disable all test and development activities.

Managing a Mix of Large and Small IOs on Exadata

Smart Scan is a core feature of Exadata, providing the ability to recognize a request to read an entire large table and to perform this work as fast as possible by parallelizing effort across the Exadata storage grid. The read requests are sent as "large IOs" so more data is collected on each read. The requests are fanned out to all the storage cells, then to all the disks, at one time. This works brilliantly for data warehouse applications, and is an excellent tool for speeding up large table scans. But if left unchecked on systems hosting mixed workloads, for example both data warehouse-like large scans and small I/O requests typical of OLTP applications, the optimizations implemented in Exadata storage for large scans could impact OLTP performance. Enter Exadata I/O Resource Manager (IORM), the Exadata feature that allows administrators to define in advance how I/O resources are used by different workloads to optimize performance in consolidated environments.

I/O Resource Manager (IORM)

Typically, IORM is configured by defining the percentage of available I/O resources each service can have. A robust E-Business Suite implementation like the customer's is basically a consolidated install, with dozens or hundreds of applications, each serving a variety of users – online and batch, with some batch programs requiring quick OLTP response times. It is not possible to define a manageable set of services to which we can assign fixed percentages for IO usage. Instead, we want to manage usage of available capacity at the storage level.

IORM has five objectives, called Basic, Balanced, Low Latency, High Throughput, and Auto.

- 'Basic' turns off IORM functionality.
- 'Low Latency' limits IOPS consumed by large IOs to about half the available capacity, plus it gives priority to small IOs if there are any queued up at the disk level. This setting is for situations where the system is very focused on rapid delivery of small IOs.
- 'Balanced' limits IOPS consumed by large IOs to a maximum of about 90% available capacity, plus it gives priority to small IOs if there are any queued up at the disk level. This setting is for situations where Exadata expects to serve small IOs / typical OLTP activity as well as needing to do some data analysis requiring table scans.
- 'High Throughput' gives priority to Smart Scans, and is for pure data warehouse environments.
- 'Auto' tells IORM to determine the optimization objective continuously and dynamically based on the workloads observed and resource plans enabled.

The customer implemented the IORM 'balanced' objective. This allows Smart Scans to be executed relatively unconstrained, thus allowing the customer to benefit from the feature. By capping Smart Scans' large IOs to slightly less than available capacity, they ensure a stream of large IO requests can be interrupted when small IO requests arrive. Then, if there are any IOs queued on the disks, priority is given to the small IOs. With this in place, small IOs are serviced promptly so the online and smaller batch processes proceed relatively unaffected while remaining unused IO capacity serves large IOs / Smart Scans. The result is minimal impact on OLTP functionality while providing significant performance improvements for larger batch jobs that scan tables.

While this is attractive for data warehouse-style work as well as some month-end processing scenarios, it is also a blessing when there is a performance bug in an application. The first full day after go-live, an unexpected data condition resulted in a custom shop floor subsystem repeatedly scanning sizable tables for several hours – a performance issue in custom code. Had Smart Scans been unrestrained, the system overall would have been overwhelmed – all users would have been negatively affected. Had Smart Scans been disabled, the shop floor subsystem would have basically hung, though the rest of the users would have seen no impact. With Smart Scans enabled and IORM 'balanced' in place, *all* users were able to continue to operate including the shop floor, while administrators identified and corrected the

performance issue by first "keeping" the pummeled table in flash cache, then by fixing the performance issues in the code.

Managing Oracle Recovery Manager (RMAN)

The customer uses RMAN to back up their databases and backs the production database up in both the production and DR data centers so there is always a local backup available for fast restore if needed. The requirement is to perform a level 0 or level 1 backup in less than four hours and maintain acceptable production performance.

The IOs issued by RMAN are large IOs, but by default are not managed directly by IORM. The customer did the following to allow IORM to identify and manage these large IOs using the same paradigm as described above:

- Disabled Database Resource Manager's (DBRM's) default_maintenance_plan.
- Created a database resource plan called RMAN_THROTTLE and set it as the default plan, which limits CPU used for backups but primarily establishes the BACKUP_GROUP.
- Enabled IORM on the storage cells, with the objective 'balanced' as described earlier.
- Created database services for backups, for each database.
- Modified their RMAN script to stop and start the services created, and to use the services for each backup channel.

Sample scripts can be found in the Appendix for these actions.

Consolidating DR with Dev/Test/QA and Performance

The customer's production data center is in the Northeast United States. They have a second data center over 600 miles away in the Southeast that hosts servers to run the business if disaster strikes the production data center. The requirements for the Disaster Recovery (DR) site are:

- Support at least 80% of the production workload in after a Data Guard failover (unplanned event) or switchover (planned event). The customer has a prioritized list of functionality to trim, to manually reduce the load to 80% if required.
- Meet a Recovery Point Objective (RPO) of a maximum 30 second data loss.
- Meet a Recovery Time Objective (RTO) of a maximum of 24 hours to fail over and bring up the application in case of a disaster.
- Mimic the production environment's configuration for testing infrastructure patches.

To meet these requirements, another Exadata X2-8 was installed in the Disaster Recovery data center. This Exadata system hosts the Data Guard physical standby replicas of the two

production databases. The disks on this install are configured with ASM high redundancy, the same as in production, so online cell maintenance can occur with reduced risk.

While this server is sized to duplicate production capacity, the standby database's redo apply process is significantly more light weight than production processing. Thus, most of the time it has an excess of CPU, memory, and IOPS capacity. To leverage this additional capacity and increase the benefits realized from the standby system, the customer also uses the environment to host all the development, test, quality assurance, and performance testing environments for the two databases. This adds two more requirements for the environment:

- Host five test environments, including two that have a physical standby – for a total of eight copies of the production databases (standby databases for production DR protection and five sets of test databases, two of which have test DR standby databases)
- Handle a performance test at 50% production capacity while the standby continues to maintain synchronization with its primary.

The customer configured high capacity disks for the DR Exadata system and added a full Exadata Storage Expansion Rack in order to provide space for these additional database copies. The storage expansion rack provides the additional benefit of increased disk IOPS to meet the customer's performance requirements.

Disk Configuration

IO using extents stored on the outer edges of a disk perform better than those using extents closer to the center of the writable space on the disks. To provide the best possible performance when production runs on the standby and to ensure the performance testing / QA database performs well, the customer configured the disks so these two databases' extents are always on the outer edges of all disks, and are spread across all available disks. Six disk groups are configured on the DR server:

ORDER OF DEFINITION	LOCATION	NAME	FUNCTION
First	Outer edge	DATA_DG	Production standby and QA/Performance tablespaces
Second	Outer	RECO_DG	Production standby and QA/Performance RECO space
Third	Outer middle	DATA_TST_DG	Dev/test tablespaces
Fourth	Inner middle	RECO_TST_DG	Dev/test RECO space
Fifth	Inner	DBFS_DG	DBFS (database file system)
Sixth	Inner edge	BKUP_DG	Space for backups to be used to refresh test databases

TABLE 2 – EXADATA DATABASE MACHINE PRODUCTION ARCHITECTURE

Resource Management

In production, the customer uses IORM to be sure that small IOs have priority over large IOs, while still allowing Smart Scans to be performed.

At the DR site, the customer needs to do this plus ensure the DR databases have all the resources they need to maintain synchronization with their primaries, followed by the QA database, and finally followed by the rest of the dev/test databases hosted on the standby system.

IORM allows "nesting" of resource definitions, where tiers can be defined, and allocations within each tier can sum to 100%. The customer used this feature to set up tiers of resource usage on the standby systems:

- Level 0 is always 100% for system requirements. It is there by default and does not need to be defined.
- If Exadata is being used to run production, the production databases will have 100% of the available resources. This is the first defined tier, Level 1 below.
- If MRP is running (the Data Guard apply process), it has priority over all other work (by definition if MRP is running a disaster has not been declared, and production is still running in the primary data center). For standby databases, each database needs a separate allocation. The customer specified 75% for the OLTP database and 25% for the Advanced Planning database – both values higher than would ever be needed by the resource. This is the second defined tier, Level 2 below.
- After MRP is served, the rest of the databases can have 100% of the remaining resources, making up the third tier. The expectation is that if the customer is running a performance test, all other test databases are shut down and will not consume any resources. This tier is also enabled by default so is not specified in the new plan.

Level 0	Level 1	Level 2	Level 3
Sys: 100%			
	Prod A, Prod B, role=primary, 100%		
		Prod A 75%, Prod B 25%, role=standby	
			"Other" 100%

Please see the sample scripts in Appendix B for creation of this plan.

Data Guard and Network Configuration

The customer's requirements for disaster recovery protection are:

- Meet an RPO of 30 seconds or better unless outside circumstances make this impossible to achieve (e.g., a double failure scenario where a primary site outage follows an earlier degradation in network connectivity that prevents the standby from being synchronized with the primary).
- Support a complete site RTO of 24 hours or better. Note that database failover occurs very quickly but that full site RTO is dependent upon restart of application tier and all ancillary systems.
- Do not affect production performance or availability.
- Provide the ability to quickly retrieve data from up to 4 hours prior, to facilitate recovery from logical data corruptions caused by operator error or program bugs.

To meet these requirements, two standby databases were deployed, as described above. For both standby databases:

- Maximum Performance Mode was configured. With this mode, primary database transactions are allowed to commit as soon as the redo is successfully written to the local online redo log, so there is no impact on production performance. A Data Guard process transmits this redo asynchronously to the standby database directly from the primary log buffer in parallel with the local log file write.
- Standby Redo Logs (SRLs) were configured in each standby database. A Data Guard process running on the standby receives the redo transmitted by the primary database and persists it on-disk in the SRL.
- At the OS level, `net.core.rmem_max`, `net.core.wmem_max`, `tcp_wmem`, and `tcp_rmem` were all set to 10 MB. The OracleNet parameters `send_buf_size` and `recv_buf_size` in both TNSNAMES and LISTENER configuration files were set to 10 MB for the connections Data Guard uses for redo transport. Send/receive buffer sizes were set to 10MB since this exceeded the value of 3xBandwidth Delay Product (BDP). BDP is the product of network bandwidth x network round-trip time. Setting send/receive buffers to 3xBDP, or min of 10MB, enables the Data Guard streaming network protocol to utilize all available bandwidth without being impacted by network latency.

While the initial requirement for the local standby was to provide a copy of the database that is 4 hours old, the customer found that recovering a four-hour backlog of redo was so fast it was easy to increase the delay to six hours to allow more time to react to a logical corruption. Thus, redo destined for the local standby is tagged to be applied with a 6-hour delay.

If a logical corruption or any other event causes data to be lost in the production database, the local standby can easily be used to research the issue and repair the problem. In such cases

MRP is stopped and the standby database is manually recovered to the appropriate point in time before the error occurred. The database is then opened as a Data Guard Snapshot Standby (this provides read-write access to the standby database), research conducted, and the data needed to repair the primary is retrieved. Once the repair is complete at the primary database and service is restored, the snapshot standby can be easily converted back into a synchronized copy of production. While the standby database functioned as a snapshot standby, it continued to receive and archive, but not apply, redo received from the primary. A single command discards any changes that were made to the snapshot standby and resynchronizes it with the primary by applying the archived redo that was sent by the primary (using the same 6-hour delay policy originally configured).

Conclusion

With the Oracle Exadata Database Machine as a foundation and Oracle's Maximum Availability Architecture as a guide, the customer built a robust solution that protects against logical and physical failures, provides outstanding performance, and accomplishes the customer's hallmark 8-hour close with room to spare. Be fast, be available, be cost-effective – The customer met these challenges with the Exadata Database Machine.

Keys to success were intelligently managing resources using IORM and DBRM so full advantage could be taken of all Exadata's features in a highly mixed workload environment, paying attention to how the data is spread across the disk farm, and using Data Guard to protect against multiple failure scenarios.

Performing a major upgrade of every component of a massive mission-critical production environment might be considered a risky event. For the first Exadata month-end, the war rooms were staffed to make sure there would be no delay in getting the right resources engaged immediately if there was an issue. The Exadata performed flawlessly and the emergencies did not occur.

Appendix A – References

An understanding of the following technology white papers and acronyms will provide the reader of this paper with a basic technical foundation.

Technical White Papers

- Oracle Exadata Database Machine: http://www.oracle.com/technetwork/database/exadata/exadata-technical-whitepaper-134575.pdf
- Exadata Smart Flash Cache Features and the Oracle Exadata Database Machine http://www.oracle.com/technetwork/server-storage/engineered-systems/exadata/exadata-smart-flash-cache-366203.pdf
- Oracle Real Application Clusters (RAC) 11g Release 2: http://www.oracle.com/technetwork/database/clustering/overview/twp-rac11gr2-134105.pdf
- Oracle Data Guard: Disaster Recovery for Exadata Database Machine http://www.oracle.com/technetwork/database/features/availability/maa-wp-dr-dbm-130065.pdf
- Deploying Oracle MAA with Exadata Database Machine: http://www.oracle.com/technetwork/database/features/availability/exadata-maa-131903.pdf

My Oracle Support Notes

- HugePages on Linux: What It Is... and What It Is Not... https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=361323.1
- Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=401749.1
- Database Machine and Exadata Storage Server 11g Release 2 (11.2) Supported Versions https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=888828.1
- Database Initialization Parameters for Oracle E-Business Suite Release 12 https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=396009.1

Acronyms

- ASCP = Advanced Supply Chain Processing
- ASM = Automatic Storage Management
- ASR = Automated Service Request
- AWR = Automatic Workload Repository
- DBFS = Database File System
- DBRM = Database Resource Manager
- DR = Disaster Recovery

- HA = High Availability
- IB = Infiniband
- IOPS = I/Os Per Second
- IORM = IO Resource Manager
- MAA = Maximum Availability Architecture
- MRP = Managed Recovery Processing or Material Requirements Planning, depending on context
- OEM = Oracle Enterprise Manager
- OLTP = OnLine Transaction Processing
- RAC = Real Application Clusters
- RMAN = Recovery Manager
- RPO = Recovery Point Objective
- RTO = Recovery Time Objective
- SGA = Shared Global Area
- SRL = Standby Redo Log

Appendix B – Sample Scripts

This section includes sample scripts to show in more detail how to configure IORM and DBRM for sharing resources on an Exadata install that serves both OLTP and data warehouse style activity. With these features configured, Smart Scans and RMAN backups can safely coexist with OLTP.

Overview

On each Exadata server, and for each database, the customer:

- Enabled and configured DBRM and IORM on the database and cells using scripts like the ones below
- Adjusted their RMAN backup script to be similar to the one below, and tested thoroughly to be sure they were configuring a proper balance between desired speed of backup and usage of resources
- Disabled the default_maintenance_plan database resource plan that is enabled automatically during default database creation.
- Applied RMAN / database patches 12811198 and 13355365 to manage their backup script across NUMA nodes in the X2-8.
    - Patch 12811198 is included in Exadata 11.2.0.3 BP8, BP11, and BP17, and in 11.2.0.4.
    - Patch 13355365 is included in Exadata 11.2.0.3 BP8, BP11, BP12, and BP17, and in 11.2.0.4.

One other thing the customer did was set an underscore parameter in their cellinit.ora files, as described below. As of their software level, IORM's directives take effect when disk queues are non-zero and have work from more than one resource. Since the customer dedicated their Exadata Database Machine to the E-Business Suite, it was possible that they needed IORM to kick in the 'balanced' objective behavior even with only one resource sending work. Please check with Support to verify you need this underscore parameter.

Implementation Steps

1. Optional: only for customers who need IORM to manage large and small IOs using the 'balanced' objective even if only one resource is sending work to the cells:
   Set the cell event that instructs IORM to honor the objective even if only one resource is sending work to the cells:

   First, back up the cellinit.ora files, using a command similar to this:

   ```
   #dcli –g ~/cell_group –l root “cp –p \$OSSCONF/cellinit.ora
   \$OSSCONF/cellinit.ora.save”
   ```

Then add the parameter _cell_iorm_wl_mode to the cellinit.ora files on the cells:

```
#dcli –g ~cell_group –l root “echo _cell_iorm_wl_mode=8 \# Make
IORM objective take effect even if only 1 resource sends work |
tee –a \$OSSCONF/cellinit.ora”
```

2. Create the IORM plan on all the cells. First create a shell script that consists of the following lines

```
cellcli -e "alter iormplan objective=’balanced’"
cellcli -e "alter iormplan catplan=''"
cellcli -e "list iormplan detail"
```

Execute this shell script (called x.sh) on all the cells using dcli. Change the “-l root” to reflect the trusted cell user (either root or celladmin). Note you cannot use cellmonitor for this.

```
$ dcli -g ~/cell_group -l root -x x.sh
```

3. Optional depending on installed version: Download and install the database patches 12811198 and 13355365 in addition to any other patches being installed.

4. Disable the scheduler windows used by the DEFAULT_MAINTENANCE_PLAN. This is required so that the RMAN_THROTTLE plan does not get disabled automatically when the maintenance jobs are scheduled.

Connect to the database as user sys. Using sysdba privilege, run:

```
exec dbms_scheduler.disable(name => 'MONDAY_WINDOW', force
=> true);
exec dbms_scheduler.disable(name => 'TUESDAY_WINDOW', force
=> true);
exec dbms_scheduler.disable(name => 'WEDNESDAY_WINDOW',
force => true);
exec dbms_scheduler.disable(name => 'THURSDAY_WINDOW', force
=> true);
exec dbms_scheduler.disable(name => 'FRIDAY_WINDOW', force
=> true);
exec dbms_scheduler.disable(name => 'SATURDAY_WINDOW', force
=> true);
exec dbms_scheduler.disable(name => 'SUNDAY_WINDOW', force
=> true);
exec dbms_scheduler.disable(name => 'WEEKEND_WINDOW', force
=> true);
exec dbms_scheduler.disable(name => 'WEEKNIGHT_WINDOW',
force => true);
```

```
select window_name,enabled from dba_scheduler_windows;
```

5. Create the RMAN_THROTTLE Resource Plan and set it as the default plan by running the script dbrmplan.sql (see below). When the script is completed the resource_manager_plan will be set to RMAN_THROTTLE.

6. Create database services for RMAN activity, to be managed by clusterware for each database:

```
srvctl add service -d qs -s qs_bkup1 -r qs1 -a qs2
srvctl add service -d qs -s qs_bkup2 -r qs2 -a qs1
```

   The naming convention for the service is <DB_NAME>_bkup[1-2] and the “-r” & “-a” are for the instance names of the database.

7. Modify the RMAN wrapper script to stop and start the above two services

   RMAN Services like those created above do not migrate back to the preferred instance automatically. This might be necessary when one instance is shutdown or otherwise closes for some unexpected reason. To ensure even distribution, the RMAN Service must be stopped and started before each backup is run.

```
srvctl stop service -d qs -s qs_bkup1
srvctl stop service -d qs -s qs_bkup2
srvctl start service -d qs -s qs_bkup1
srvctl start service -d qs -s qs_bkup2
```

   This is tracked under bug# 14583029 - SERVICES STARTED ON WRONG NODE DURING SRVCTL START DATABASE

8. The RMAN Script should be configured similar to the following. The customer changed the “allocate channel” syntax to reflect that needed by Symantec NetBackup

```
 CONFIGURE COMPRESSION ALGORITHM clear;

run

{
sql 'alter system set "_backup_disk_bufcnt"=64
scope=memory';
sql 'alter system set "_backup_disk_bufsz"=1048576
scope=memory';
sql 'alter system set "_backup_file_bufcnt"=64
scope=memory';
sql 'alter system set "_backup_file_bufsz"=1048576
scope=memory';
```

```
allocate channel ch01 device type disk connect
'sys/welcome1@<scan_name>/qs_bkup1' format
'/zfssa/qs/backup1/%U';
allocate channel ch10 device type disk connect
'sys/welcome1@<scan_name>/qs_bkup2' format
'/zfssa/qs/backup4/%U';

backup as backupset incremental level 0 section size 64g
filesperset 16 database tag 'level0';
}
```

Notes on the above:

- The first line disabled RMAN Compression (note: the customer has not purchased Advanced Compression)
- The 4 “sql” commands will override the default number of buffers and channels used to read from the database. By default RMAN will allocate 4MB buffers, and the number of buffers total the number of disks in the ASM Disk Group.
- The Allocate Channel commands will be modified to reflect those required by Symantec NetBackup. However, the “connect” states who to connect as and which instance to connect to. We want to allocate 1 channel to each instance. Use the service name from step 5 above
- The backup command is self explanatory. We will be performing an RMAN Backup, creating an incremental level 0 backupset. If there are any “bigfile” tablespaces in the database greater than 64g in size, we will back them up in 64g pieces. Otherwise we will put 16 data files in each backup piece. We are performing a full database backup, and are giving the backup a tag ‘Level0’;

9. The RMAN script for the nightly level 1 backups will be identical to that of the level 0 backup script above, with the exception that level 1 will be stated as follows

```
CONFIGURE COMPRESSION ALGORITHM clear;

run

{
sql 'alter system set "_backup_disk_bufcnt"=64 scope=memory';
sql 'alter system set "_backup_disk_bufsz"=1048576
scope=memory';
sql 'alter system set "_backup_file_bufcnt"=64 scope=memory';
sql 'alter system set "_backup_file_bufsz"=1048576
scope=memory';

allocate channel ch01 device type disk connect
'sys/welcome1@<scan_name>/qs_bkup1' format
'/zfssa/qs/backup1/%U';
allocate channel ch10 device type disk connect
'sys/welcome1@<scan_name>/qs_bkup2' format
'/zfssa/qs/backup4/%U';
```

```
backup as backupset incremental level 1 section size 64g
filesperset 16 database tag 'level1';
}
```

dbrmplan.sql:

This script creates the RMAN_THROTTLE DBRM plan needed to isolate RMAN's large writes so IORM can properly manage them.

```
begin
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA;
  DBMS_RESOURCE_MANAGER.SWITCH_PLAN( plan_name => '');
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA;
end;
/

begin
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA;
  DBMS_RESOURCE_MANAGER.DELETE_PLAN( plan => 'RMAN_THROTTLE');
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA;
end;
/

begin
  DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA;
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA;
  DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING( attribute
=> DBMS_RESOURCE_MANAGER.ORACLE_FUNCTION, value => 'BACKUP',
consumer_group => 'BACKUP_GROUP');
  DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING( attribute
=> DBMS_RESOURCE_MANAGER.ORACLE_FUNCTION, value => 'COPY',
consumer_group => 'BACKUP_GROUP');
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA;
end;
/

begin
  DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP(
grantee_name => 'PUBLIC', consumer_group => 'BACKUP_GROUP',
grant_option => TRUE );
end;
/


begin
  DBMS_RESOURCE_MANAGER.CLEAR_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.CREATE_PLAN( plan => 'RMAN_THROTTLE',
```

```
    comment => 'RESOURCE PLAN FOR RMAN THROTTLE DOWN OF
RESOURCES');

  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(
    plan => 'RMAN_THROTTLE',
    group_or_subplan => 'BACKUP_GROUP',
    comment => 'LOW PRIORITY FOR RMAN BACKUP OPERATIONS',
    mgmt_p1 => 20,
    max_utilization_limit => 60);

  DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE(
    plan => 'RMAN_THROTTLE',
    group_or_subplan => 'OTHER_GROUPS',
    comment => 'PROCESSING FOR THE REST',
    mgmt_p2 => 100);
  DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/

begin
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA;
  DBMS_RESOURCE_MANAGER.SWITCH_PLAN( plan_name =>
'RMAN_THROTTLE');
  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA;
end;
/
```

Oracle Exadata Technical Case Study:
March, 2015
Author: Lyn Pratt, with great assistance from the Oracle High Availability Product Management and MAA Team Members

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together