ARTICLE

Going Live On Oracle Exadata

Marc Fielding - Senior Consultant

Pythian
love your data

This is the story of a real-world Exadata Database Machine deployment integrating OBIEE analytics and third-party ETL tools in a geographically distributed, high-availability architecture. Learn about our experiences with large-scale data migration, hybrid columnar compression and overcoming challenges with system performance. Find out how Exadata improved response times while reducing power usage, data center footprint and operational complexity.

The Problem

LinkShare provides marketing services for some of the world's largest retailers, specializing in affiliate marketing, lead generation, and search[1]. LinkShare's proprietary Synergy Analytics platform gives advertisers and website owners real-time access to online visitor and sales data, helping them manage and optimize online marketing campaigns. Since the launch of Synergy Analytics, request volumes have grown by a factor of 10, consequently putting a strain on the previous database infrastructure.

This strain manifested itself not only in slower response times, but also increasing difficulty in maintaining real-time data updates, increased database downtime and insufficient capacity to add large clients to the system. From the IT perspective, the legacy system was nearing its planned end-of-life replacement period. Additionally, monthly hard disk failures would impact performance system-wide as data was rebuilt onto hot spare drives. I/O volumes and storage capacity were nearing limits and power limitations in the datacenter facilities made it virtually impossible to add capacity to the existing system. Therefore, the previous system required a complete replacement.

The Solution

The end-of-life of the previous system gave an opportunity to explore a wide range of replacement alternatives. They included a newer version of the legacy database system, a data warehouse system based on Google's MapReduce[2] data-processing framework and Oracle's Exadata database machine. Ultimately, Exadata was chosen as the replacement platform for a variety of factors, including the superior failover capabilities of Oracle RAC and simple, linear scaling that the Exadata architecture provides. It was also able to fit in a single rack what had previously required three racks, along with an 8x reduction in power usage. Exadata was able to deliver cost savings and improved coverage by allowing the same DBAs that manage the existing Oracle-based systems to manage Exadata as well. Once Exadata hardware arrived, initial installation and configuration was very fast, assured with a combination of teams from implementation partner Pythian; Oracle's strategic customer program, Oracle Advanced Customer Services; and LinkShare's own DBA team. In less than a week, hardware and software was installed and running. The Architecture User requests are handled through a global load balancing infrastructure, able to balance loads across datacenters and web servers. A cluster of web servers and application servers run Oracle Business Intelligence Enterprise Edition (OBIEE), a business intelligence tool allowing users to gain insight into online visitor and sale data from a familiar web browser interface. The OBIEE application servers are then connected to an Exadata database machine.

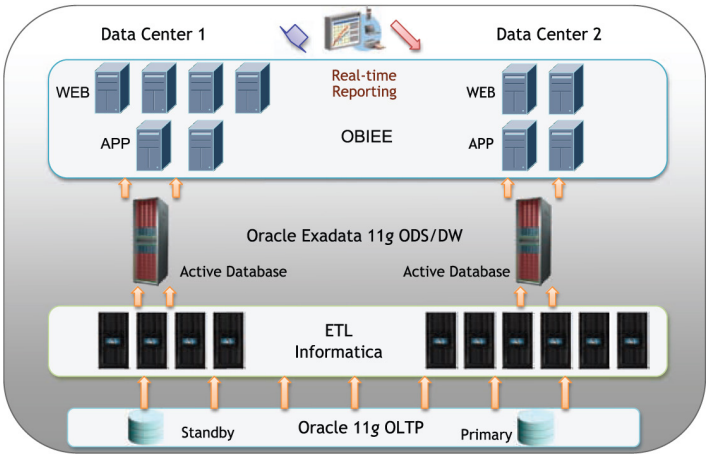

Figure 1. Overall System Architecture

Data flows from Oracle 11g-based OLTP systems, using a cluster of ETL servers running Informatica PowerCenter that extract and transform data for loading into an operational data store (ODS) schema located on the Exadata system. The ETL servers then take the ODS data,

[1] *Affiliate Programs - LinkShare http://www.linkshare.com*
[2] *MapReduce: Simplified Data Processing on Large Clusters, Jeffrey Dean, Sanjay Ghemawat. http://labs.google.com/papers/mapreduce-osdi04.pdf*

further transforming it into a dimensional model in a star schema. The star schema is designed for flexible and efficient querying as well as storage space efficiency.

LinkShare's analytics platform serves a worldwide client base and doesn't have the off-hours maintenance windows common to many other analytics systems. The high availability requirements dictated an architecture (Fig. 1) that relies not on the internal redundancy built into the Exadata platform, but also to house two independent Exadata machines in geographically separated datacenter facilities. Rather than using a traditional Oracle Data Guard configuration, LinkShare opted to take advantage of the read-intensive nature of the analytics application to simply double-load data from source systems using the existing ETL platform. This configuration completely removes dependencies between sites and also permits both sites to service active users concurrently.

In order to reduce migration risks and to permit an accelerated project timeline, application and data model changes were kept to a bare minimum. The largest application code changes involved handling differences in date manipulation syntax between Oracle and the legacy system. The logical data model, including ODS environment and star schema, was retained.

The legacy system had a fixed and inflexible data partitioning scheme as a by-product of its massively parallel architecture. It supported only two types of tables: nonpartitioned tables, and partitioned tables using a single numeric partition key, hashed across data nodes. The requirement to have equal-sized partitions to maintain performance required the creation of a numeric incrementing surrogate key as both primary key and partition key. The move to Oracle opened up a whole new set of partitioning possibilities that better fit data access patterns, all with little or no application code changes. More flexible partitioning allows improved query performance, especially when combined with full scans, as well as simplifying maintenance activities like the periodic rebuild and recompression of old data. The final partition layout ended up combining date range-based partitioning with hash-based subpartitioning on commonly queried columns.

Data Migration

Data migration was done in three separate ways, depending on the size of the underlying tables. Small tables (less than 500MB in size) were migrated using Oracle SQL Developer's built-in migration tool. This tool's GUI interface allowed ETL developers to define migration rules independently of the DBA team, freeing up DBA time for other tasks. Data transfer for these migrations was done through the developers' own desktop computers and JDBC drivers — on a relatively slow network link — so these transfers were restricted to small objects. The table definitions and data were loaded into a staging schema, allowing them to be examined for correctness by QA and DBA teams before being moved in bulk to their permanent location.

Larger objects were copied using existing Informatica PowerCenter infrastructure and the largest objects (more than 10GB) were dumped to text files on an NFS mount using the legacy system's native query tools, and loaded into the Exadata database using SQL*Loader direct path loads. Simultaneous parallel loads on different partitions improved throughput. Initial SQL*Loader scripts were generated from Oracle SQL Developer's migration tool but were edited to add the UNRECOVERABLE, PARALLEL and PARTITION keywords, enabling direct path parallel loads. The SQL*Loader method proved to be more than twice as fast as any other migration method, so many of the tables originally planned to be migrated by the ETL tool were done by SQL*Loader instead. (Although SQL*Loader was used here because of DBA team familiarity, external tables are another high-performance method of importing text data.)

Another tool commonly used in cross-platform migrations is Oracle Transparent Gateways. Transparent gateways allow non-Oracle systems to be accessed through familiar database link interfaces as if they were Oracle systems. We ended up not pursuing this option to avoid any risk of impacting the former production environment, and to avoid additional license costs for a short migration period.

One of the biggest challenges in migrating data in a 24x7 environment is not the actual data transfer; rather, it is maintaining data consistency between source and destination systems without incurring downtime. We addressed this issue by leveraging our existing ETL infrastructure: creating bidirectional mappings for each table and using the ETL system's

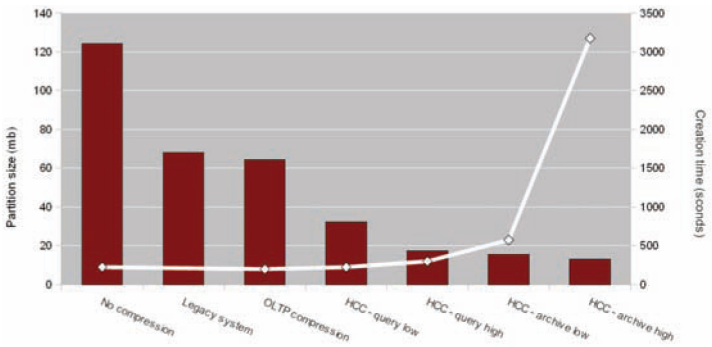

Figure 2: Comparison of Compression Rates

change-tracking capabilities to propagate data changes made in either source or destination system. This process allowed the ETL system to keep data in the Exadata systems up to date throughout the migration process. The process was retained post-migration, keeping data in the legacy system up to date

One of Exadata's headline features is hybrid column compression, which combines columnar storage with traditional data compression algorithms like LZW to give higher compression ratios than traditional Oracle data compression. One decision when implementing columnar compression is choosing acompression level; the compression levels between QUERY LOW and ARCHIVE HIGH offer increasing tradeoffs between space savings and compression overhead.[3] Using a sample table to compare compression levels (Fig. 2), we found the query high compression level to be at the point of diminishing returns for space savings, while still offering competitive compression overhead. In the initial implementation, a handful of large and infrequently accessed table partitions were compressed with hybrid columnar compression, with the remaining tables using OLTP compression. Based on the good results with columnar compression, however, we plan to compress additional tables with columnar compression to achieve further space savings.

Performance Tuning

Avoiding Indexes

Improving performance was a major reason for migrating to Exadata and made up a large part of the effort in the implementation project. To make maximum use of Exadata's offload functionality for the data-intensive business intelligence workload, it was initially configured with all indexes removed. (This approach would not be recommended for workloads involving online transaction processing, however.) The only the exceptions were primary key indexes required to avoid duplicate rows, and even these indexes were marked as INVISIBLE to avoid their use in query plans. Foreign key enforcement was done at the ETL level rather than inside the database, avoiding the need for additional foreign key indexes. By removing or hiding all indexes, Oracle's optimizer is forced to use full scans. This may seem counterintuitive; full scans require queries to entire table partitions, as compared to an index scan, which reads only the rows matching query predicates. But by avoiding index scans, Exadata's smart scan storage offloading capability can be brought to bear. Such offloaded operations run inside Exadata storage servers, which can use their directly attached disk storage to efficiently scan large volumes of data in parallel. These smart scans avoid one of the major points of contention with rotating storage in a database context: slow seek times inherent in single-block random I/O endemic in index scans and ROWID-based table lookups.

Exadata storage servers have optimizations to reduce the amount of raw disk I/O. Storage indexes cache high and low values for each storage region, allowing I/O to be skipped entirely when there is no possibility of a match. The Exadata smart flash cache uses flash-based storage to cache the most frequently used data, avoiding disk I/O if data is cached. The net result is that reading entire tables can end up being faster than traditional index access, especially when doing large data manipulations common in data warehouses like LinkShare's.

Benchmarking Performance

Given the radical changes between Exadata and the legacy environment, performance benchmarks were essential to determine the ability of the Exadata platform to handle current and future workload. Given that the Exadata system had less than 25 percent of the raw disk spindles and therefore less I/O capacity compared to the legacy system, business management was concerned that Exadata performance would degrade sharply under load.

To address these concerns, the implementation team set up a benchmark environment where the system's behavior under load could be tested. While Oracle-to-Oracle migrations may use Real Application Testing (RAT) to gather workloads and replay them

[3] *Oracle Database Concepts11g Release 2 (11.2)*

performance testing, RAT does not support on-Oracle platforms. Other replay tools involving Oracle trace file capture were likewise not possible.

Eventually a benchmark was set up at the webserver level using the opensource JMeter[4] tool to read existing webserver logs from the legacy production environment and reformat them into time-synchronized, simultaneous requests to a webserver and application stack connected to the Exadata system. This approach had a number of advantages, including completely avoiding impacts to the legacy environment and using testing infrastructure with which the infrastructure team was already familiar. A side benefit of using playback through a full application stack was that it allowed OBIEE and web layers to be tested for performance and errors. Careful examination of OBIEE error logs uncovered migration-related issues with report structure and query syntax that could be corrected. Load replay was also simplified by the read-intensive nature of the application, avoiding the need for flashback or other tools to exactly synchronize the database content with the original capture time.

The benchmark was first run with a very small load — approximately 10 percent of the rate of production traffic. At this low rate of query volume, overall response time was about 20 percent faster than the legacy system. This was a disappointment when compared to the order of magnitude improvements expected, but it was still an improvement.

The benchmark load was gradually increased to 100 percent of production volume. Response time slowed down dramatically to the point where the benchmark was not even able to complete successfully. Using database-level performance tools like Oracle's AWR and SQL monitor, the large smart scans were immediately visible, representing the majority of response time.

Another interesting wait event was visible: enq: KO – fast object checkpoint. These KO waits are a side effect of direct-path reads, including Exadata smart scans. Another session was making data changes — in this case updating a row value. But such updates are buffered and not direct-path, so they are initially made to the in-memory buffer cache only. But direct-path reads, which bypass the buffer cache and read directly from disk, wouldn't see these changes. To make sure data is consistent, Oracle introduces the enq: KO – fast object checkpoint wait event, waiting for the updated blocks to be written to disk. The net effect is that disk reads would hang, sometime for long periods of time, until block checkpoints could complete. Enq: KO – fast object checkpoint waits can be avoided by doing direct-path data modifications. Such data changes apply only to initially empty blocks, and once the transaction is committed, the changed data is already made on disk. Unfortunately, direct-path data modifications can only be applied to bulk inserts using the /*+APPEND*/ hint or CREATE TABLE AS SELECT, not UPDATE or DELETE.

Operating system level analysis on the storage servers using the Linux iostat tool showed that the physical disk drives were achieving high read throughput and running at 100 percent utilization, indicating that the hardware was functioning properly but struggling with the I/O demands placed on it.

Solving the Problem

To deal with the initial slow performance, we adopted a more traditional data warehousing feature of Oracle: bitmap indexes and star transformations.[5] Bitmap indexes work very differently from Exadata storage offload, doing data processing at the database server level rather than offloading to Exadata storage servers. By doing index-based computations in advance of fact table access, they only retrieve matching rows from fact tables. Fact tables are generally the largest table in a star schema, thus, bitmap-based data access typically does much less disk I/O than smart scans, at the expense of CPU time, disk seek time, and reduced parallelism of operations. By moving to bitmap indexes, we also give up Exadata processing offload, storage indexes and even partition pruning, because partition join filters don't currently work with bitmap indexes. With the star schema in place at LinkShare, however, bitmap indexes on the large fact tables allowed very efficient joins of criteria from dimension tables, along with caching benefits of the database buffer cache. The inherent space efficiencies of bitmap indexes allowed aggregatete index size to remain less than 30 percent of the size under the legacy system.

Query-Level Tuning

Even with bitmap indexes in place, AWR reports from benchmark runs identified a handful of queries with unexpectedly high ratios of logical reads per execution.

4 Apache JMeter http://jakarta.apache.org/jmeter/

A closer look at query plans showed the optimizer dramatically underestimating row cardinality, and in turn choosing nested-loop joins when hash joins would have been an order of magnitude more efficient. Tuning options were somewhat limited because OBIEE's SQL layer does not allow optimizer hints to be added easily. We instead looked at the SQL tuning advisor and SQL profiles that are part of Oracle Enterprise Manager's tuning pack. In some cases, the SQL tuning advisor was able to correct the row cardinality estimates directly and resolve the query issues by creating SQL profiles with the OPT_ESTIMATE query hint.[6] SQL profiles automatically insert optimizer hints whenever a given SQL statement is run, without requiring application code changes. OBIEE, like other business intelligence tools, generates SQL statements without bind variables, making it difficult to apply SQL profiles to OBIEE-generated SQL statements. A further complication came from lack of bind variables in OBIEE-generated SQL statements. Beginning in Oracle 11gR1, the FORCE_MATCH option to the DBMS_SQLTUNE.ACCEPT_SQL_PROFILE procedure[7] comes to the rescue, matching any bind variable in a similar manner than the CURSOR_SHARING=FORCE initialization parameter.

In many cases, however, the SQL tuning advisor simply recommended creating index combinations that make no sense for star transformations. In these cases, we manually did much of the work the SQL tuning advisor would normally do by identifying which optimizer hints would be required to correct the incorrect assumptions behind the problematic execution plan. We then used the undocumented DBMS_SQLTUNE.IMPORT_SQL_PROFILE function[8] to create SQL profiles that would add hints to SQL statements much the way the SQL tuning advisor would normally do automatically. Analyzing these SQL statements manually is a very time-consuming activity; fortunately, only a handful of statements required such intervention.

Going Live

LinkShare's Exadata go-live plan was designed to reduce risk by slowly switching customers from the legacy system while preserving the ability to revert should significant problems be discovered. The ETL system's simultaneous loads kept all systems up to date, allowing analytics users to run on either system. Application code was added to the initial login screen to direct users to either the legacy system or the new system based on business-driven criteria. Initially, internal users only were directed at Exadata, then 1 percent of external users, ramping up to 100 percent within two weeks. Go-live impacts on response time were immediately visible from monitoring graphs, as shown in Fig. 3. Not only did response times improve, but they also became much more consistent, avoiding the long outliers and query timeouts that would plague the legacy system.

The second data center site went live in much the same manner, using the ETL system to keep data in sync between systems and slowly ramping up traffic to be balanced between locations.

Operational Aspects

Given that Exadata has a high-speed InfiniBand network fabric, it makes sense to use this same fabric for the I/O-intensive nature of database backups. LinkShare commissioned a dedicated backup server with an InfiniBand host channel adapter connected to one of the Exadata InfiniBand switches. RMAN backs up the ASM data inside the Exadata storage servers using NFS over IP over InfiniBand. Initial tests were constrained by the I/O capacity of local disk, so storage was moved to an EMC storage area network (SAN) already in the datacenter, using the media server simply as a NFS server for the SAN storage.

Monitoring is based on Oracle Enterprise Manager Grid Control to monitor the entire Exadata infrastructure. Modules for each Exadata component, including database, cluster, Exadata storage servers, and InfiniBand hardware, give a comprehensive status view and alerting mechanism. This is combined with Foglight[9], a third-party tool already extensively used for performance trending within LinkShare, installed on the database servers. The monitoring is integrated with Pythian's remote DBA service, providing both proactive monitoring and 24x7 incident response.

Patching in Exadata involves several different layers: database software, Exadata storage servers, database-server operating system components like infiniBand

5 Oracle Database Data Warehousing Guide 11g Release 2 (11.2)
6 Oracle's OPT_ESTIMATE hint: Usage Guide, Christo Kutrovsky. http://www.pythian.com/news/13469/oracles-opt_estimate-hint-usage-guide/
7 Oracle Database Performance Tuning Guide 11g Release 2 (11.2)
8 SQL Profiles, Christian Antognini, June 2006. http://antognini.ch/papers/SQLProfiles_20060622.pdf
9 Quest Software Foglight http://www.quest.com/foglight/

Going Live On Oracle Exadata

Marc Fielding - Senior Consultant

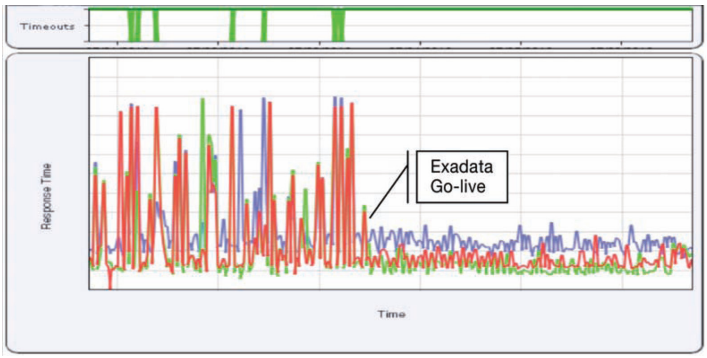

Figure 3: Monitoring-server Response Times Before and After Exadata Go-Live

drivers, and infrastructure like InfiniBand switches, ILOMv lights-out management cards in servers, and even console switches and power distribution units. Having a second site allows us to apply the dwindling number of patches that aren't rolling installable by routing all traffic to one site and installing the patch in the other.

Looking Ahead

With Exadata sites now in production, development focus is shifting to migrating the handful of supporting applications still running on the legacy system. Retirement of the legacy system has generated immediate savings in data center and vendor support costs, as well as freeing up effort in DBA, ETL and development teams to concentrate on a single platform.

On the Exadata front, the roadmap focuses on making better use of newly available functionality in both the Exadata storage servers and the Oracle platform in general. In particular, we're looking at making more use of Exadata's columnar compression, incorporating external tables into ETL processes, and making use of materialized views to precompute commonly queried data.

The Results

The move to Exadata has produced quantifiable benefits for LinkShare. Datacenter footprint and power usage have dropped by factors of 4x and 8x, respectively. The DBA team has one less platform to manage. Response times have improved by factors of 8x or more, improving customer satisfaction. The ability to see more current data has helped users make better and timelier decisions. And, ultimately, improving customer retention and new customer acquisition.

Originally published in the Q42011 issues of the IOUG Select Journal.

About The Author

Marc Fielding is senior consultant with Pythian's advanced technology group where he specializes in high availability, scalability and performance tuning. He has worked with Oracle database products throughout the past 10 years, from version 7.3 up to 11gR2. His experience across the entire enterprise application stack allows him to provide reliable, scalable, fast, creative and cost-effective solutions to Pythian's diverse client base. He blogs regularly on the Pythian blog www.pythian.com/news, and is reachable via email at fielding@pythian.com, or on twitter @pythianfielding.

About Pythian

Pythian is a global database and application infrastructure services company for Oracle, MySQL and SQL Server. Since 1997, companies have entrusted Pythian to keep their database infrastructures running efficiently and to help them strategically align their IT and business goals. Pythian's unparalleled DBA skills, mature methodologies, best practices and tools enable clients to do more with fewer resources. Pythian's corporate headquarters is in Ottawa Canada, with offices worldwide.
To find out more visit http://www.pythian.com/Exadata