Making the Most of Oracle Exadata

by Marc Fielding

Marc Fielding

At Oracle's third-quarter earnings call, Larry Ellison announced that Oracle Exadata is "well on its way to being the most successful product launch in Oracle's 30-year history," with a sales pipeline approaching $1 billion for 2011. He attributed these sales to the game-changing performance of the Oracle Exadata platform. But what is the "secret sauce" behind these performance numbers? Read on to learn about the major performance features of Oracle Exadata, and discover tips on how to maximize performance from those features based on the author's own experience implementing Exadata.

Key Features

Smart scans: Smart scans are Exadata's headline feature. They provide three main benefits: reduced data transfer volumes from storage servers to databases, CPU savings on database servers as workload is transferred to storage servers, and improved buffer cache efficiency thanks to column projection. Smart scans use helper processes that function much like parallel query processes but run directly on the storage servers. Operations off-loadable through smart scans include the following:

- **Predicate filtering**—processing WHERE clause comparisons to literals, including logical operators and most SQL functions.
- **Column projection**—by looking at a query's SELECT clause, storage servers return only the columns requested, which is a big win for wide tables.
- **Joins**—storage servers can improve join performance by using Bloom filters to recognize rows matching join criteria during the table scan phase, avoiding most of the I/O and temporary space overhead involved in the join processing.
- **Data mining model scoring**—for users of Oracle Data Mining, scoring functions like PREDICT() can be evaluated on storage servers.

Storage indexes: Storage indexes reduce disk I/O volumes by tracking high and low values in memory for each 1-megabyte storage region. They can be used to give partition pruning benefits without requiring the partition key in the WHERE clause, as long as one of these columns is correlated with the partition key. For example, if a table has order_date and processed_date columns, is partitioned on order_date, and if orders are processed within 5 days of receipt, the storage server can track which processed_date values are included in each order partition, giving partition pruning for queries referring to either order_date or processed_date. Other data sets that are physically ordered on disk, such as incrementing keys, can also benefit.

Columnar compression: Hybrid columnar compression (HCC) introduces a new physical storage concept, the compression unit. By grouping many rows together in a compression unit, and by storing only unique values within each column, HCC provides storage savings in the range of 80–90% based on the compression level selected. Since data from full table scans remains compressed through I/O and buffer cache layers, disk savings translate to reduced I/O and buffer cache work as well. HCC does, however, introduce CPU and data modification overhead that will be discussed in the next section.

"By grouping many rows together in a compression unit, and by storing only unique values within each column, HCC provides storage savings in the range of 80–90% based on the compression level selected."

Flash cache: Exadata's flash cache supplements the database servers' buffer caches by providing a large cache of 384 GB per storage server and up to 5 TB in a full Oracle Exadata Database Machine, considerably larger than the capacity of memory caches. Unlike generic caches in traditional SAN storage, the flash cache understands database-level operations, preventing large non-repeated operations such as backups and large table scans from polluting the cache. Since flash storage is nonvolatile, it can cache synchronous writes, providing performance benefits to commit-intensive applications.

Hot/cold storage: The inherent geometry of rotating disks means that data is stored more densely in the outer portion of disk platters, giving higher throughput for disk operations on outer tracks and reducing the amount of time spent on head movement. Some Oracle systems currently leave inner tracks completely unused for this reason. Exadata allows the creation of separate "hot" ASM diskgroups for performance-critical data in outer disk regions and "cold" diskgroups for fast recovery area use in the inner regions.

I/O resource manager: Exadata's I/O Resource Manager (IORM) permits disk I/O operations to be prioritized on the storage cell in the same fashion as CPU time, and parallel query processes are currently managed by the Database Resource Manager (DBRM) on the database server. IORM is particularly useful when consolidating multiple database workloads together, by allowing I/O capacity to be allocated between different workloads according to their importance and configured limits.

Balanced hardware: The Exadata-powered Oracle Database Machine includes a fixed ratio of database nodes, storage servers, and associated networking equipment designed to avoid performance bottlenecks in any single part of the infrastructure.

Optimizing Performance

Exadata's performance features have been designed to work out of the box and require no manual configuration to use. That being said, a few small optimizations can greatly improve their effectiveness, and below are some battle-tested performance tips based on real-world Exadata deployments.

Use parallel query: Each Exadata storage cell contains 12 disks, which adds up to 168 disks in a full rack configuration. A key to efficient use of disk resources is to spread workload over these disks and particularly to avoid sequential operations that must wait until one disk operation completes before starting another. The ASM storage layer uses striping to distribute data evenly across physical disks, but it is up to the database instances to send I/O requests in such a way that the disks stay busy, which is where parallel query comes in. By splitting database operations into small chunks, parallel query can keep multiple physical drives busy, thus improving query response time.

The challenge in using parallel query is that the same optimizations that improve response time at slow periods can actually reduce scalability during periods of high demand, due to contention for the hard drives' fixed I/O capacity. Additionally, too many parallel processes can overwhelm system resources on database servers, causing additional performance degradation. To combat this problem, the Oracle database sets a limit on parallel query processes, PARALLEL_MAX_SERVERS. Once the system reaches PARALLEL_MAX_SERVERS, new requests run without parallelism at all, creating high variability in response time. The best way to avoid this situation is to avoid having it happen in the first place, using the Database Resource Manager to control maximum parallelism. Based on an analysis of expected concurrency and parallel query capacity (which is typically 128 and 256 processes on Exadata database nodes, depending on the mix of full-scan and more-CPU-intensive index operations), a resource manager plan can be constructed involving limits on both the number of parallel query processes per session and the total number of concurrent sessions.

While adding concurrency is of great benefit to large table scans, the overhead of managing parallel query processes can actually slow down scans of small tables. As a rule of thumb, if table operations take less than half a second to complete, parallel query delivers diminishing benefits, although this threshold can vary based on workload patterns. It is relatively simple to benchmark by testing common queries with varying degrees of parallelism, taking into account the system's total process capacity.

"The challenge in using parallel query is that the same optimizations that improve response time at slow periods can actually reduce scalability during periods of high demand, due to contention for the hard drives' fixed I/O capacity. Additionally, too many parallel processes can overwhelm system resources on database servers, causing additional performance degradation."

Learn to love the full scan: Index-based access paths work by generating a list of unique row identifiers from an index and then looking up the rows one by one from the source table. While very efficient for small lookups, the overhead of such sequential, random disk accesses increases nearly linearly as data size increases, making full scans more efficient for the large retrievals common in data warehouse workloads.

Since index access paths perform filtering at the database server rather than the storage server, Exadata's smart scan row filtering and join filters offer little benefit. Index-based table lookup performance further degrades with columnar compression, since even a single-row lookup requires reading an entire compression unit.

Good full-scan performance hinges on a good partition layout. Careful thought needs to be given to permit full partitions to closely match the data that users typically request.

Oracle provides several tools to help:

- A wide variety of partition types, including range, hash, list, interval, reference, or even virtual columns
- Composite subpartitioning with different partition and subpartition key columns, providing further benefits for queries involving multiple columns

- ➤ Storage indexes, giving the same benefits as partition pruning when data is clustered on disk

Although matching partition layout with query data sets provides maximum benefit of Exadata's features, it's rarely possible to match a partition layout to every possible query. In such cases, indexes are still required. Referential integrity continues to require indexes as well, although primary key indexes can be set to INVISIBLE (and therefore not be considered by the optimizer) if they prevent legitimate full partition scans.

Don't stop tuning applications: The advent of Exadata does not replace the need for application-level tuning. Poorly scaling application code will not suddenly become scalable when run on Exadata. The same tuning methods used in regular Oracle RAC databases will continue to work, so take advantage of the volumes of application-level tuning resources already available for Oracle RAC platforms.

"Businesses today are faced with ever-increasing user demands and data volumes. The combination of Exadata's feature set with a well-focused performance tuning effort can help address these challenges while benefiting from the results of 20 years of Oracle product development."

Spread disk groups across all available cell disks: Exadata's storage architecture makes it easy to implement SAME (Stripe and Mirror Everything) striping across all available drives. Thanks to the I/O Resource Manager, the performance guarantees previously only available by dedicating disks to specific applications can be obtained on a SAME layout. A SAME disk layout not only allows more efficient use of space, but it also gives better utilization of I/O capacity as well. For example, production applications can be guaranteed a certain percentage of I/O capacity—or even absolute priority when they require it—leaving otherwise available capacity for non-production applications.

Manage allocation units and extents carefully: Exadata storage cells are divided up into individual allocation units managed by ASM, analogous to stripes in a RAID configuration. Increasing ASM allocation unit size from the default 1 MB to 4 MB or even 8 MB improves performance of large reads by reducing the amount of seeking that drives need to do, along with reducing the overhead for managing allocation units themselves. The table-scanning benefits of larger allocation units only happen when they can be filled with a single table or index extent. To ensure that allocation units are not shared between multiple extents, set initial extents in large tables to be at least as large as the ASM allocation unit size. This process can be automated by using the CELL_PARTITION_LARGE_EXTENTS initialization parameter, which sets initial extents to 8 MB automatically for table partitions. Large 8 MB initial extents can be wasteful when storing very small objects, however. So for objects not expected to grow, smaller initial extents are still appropriate.

Reserve the flash cache for caching: Objects can be stored on flash disks permanently by creating flash-based grid disks. However, this type of storage layout usually hurts performance, since the built-in caching logic can do a better job of identifying which exact data blocks are the most frequently used and make intelligent caching decisions based on this information. Additionally, since true cache data does not need normal redundancy, a cache can store twice as much data in the same amount of storage as a grid disk.

Consider Oracle Secure Backup: Oracle Secure Backup (OSB) runs on media servers directly connected to the Exadata InfiniBand fabric. In its current form, however, OSB has an important limitation: it can only back up to dedicated tape devices directly attached to OSB media servers. Neither disk-based backups nor third-party backup servers are supported. In environments that can accommodate this restriction, however, OSB eliminates network traffic on the database servers' Ethernet interfaces.

Use columnar compression judiciously: Hybrid columnar compression (HCC) in Exadata has the dual advantages of reducing storage usage and reducing I/O for large reads by storing data more densely. However, HCC works only when data is inserted using bulk operations. If non-compatible operations like single-row inserts or updates are attempted, Exadata reverts transparently to the less restrictive OLTP compression method, losing the compression benefits of HCC. When performing data modifications such as updates or deletes, the entire compression unit must be uncompressed and written in OLTP-compressed form, involving an additional disk I/O penalty as well. To avoid such overhead, consider compressing only data that infrequently changes, such as historical data. If partitioning data by date and data modification happens occasionally, a scripted automated process could periodically re-compress older partitions.

Conclusion

Businesses today are faced with ever-increasing user demands and data volumes. The combination of Exadata's feature set with a well-focused performance tuning effort can help address these challenges while benefiting from the results of 20 years of Oracle product development. ▲

Marc Fielding is a senior consultant with Pythian, a leading database infrastructure and application services company, where he specializes in high availability, scalability, and performance. Having worked with Oracle database products over the past nine years, 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. Read his blog posts at **www.pythian.com/news/tag/exadata** *or contact him at* **fielding@pythian.com**.