Monitoring Exadata with Precise

Oracle Exadata comprises of Oracle database servers, Oracle Exadata Storage Servers, and an InfiniBand fabric for storage networking in a single Database Machine. Oracle markets Exadata for both OLTP and BI workloads. A logical representation is shown below.

![Exadata Diagram]

Having both workloads on the same Exadata can mean you get much better utilization of the total capacity of the Exadata since each workload type favours a different part of the infrastructure that makes up Exadata. The BI workload can be Offloaded and use the CPU and Memory on the Storage Cells instead of the DB Compute Nodes. The full scans will return directly to the PGA with Offloading and this means the Buffer Cache is not affected by the BI activity. This means that the DB Compute Nodes will have much more capability to handle OLTP functionality more efficiently.

Concepts

Oracle uses a technique called “Offloading” or Smart Scan to push the processing down to the storage cells instead of being carried out on the database compute nodes. The basic concept of offloading is to make the storage system smart enough to filter out as much data as possible before returning it to the database. It reduces the amount of data returned, and returns that data over a much faster channel - Infiniband. By doing this, Exadata drastically reduces the impact of physical I/O on the database. This removes one of the major performance bottlenecks and speeds the system up significantly.

Smart Scan or Offloading runs when the query really requires a full scan to be executed using direct path reads. This means that it does not run for everything that the database executes. The optimizer must choose to do either a Full Table Scan or a Fast Full Index Scan in order for Smart Scan to be possible. That scan must be a direct path read which will allow parallel slaves to process the data on each storage cell. The database will check with ASM to see which storage cells contain that object and will fork off parallel requests to each storage cell asking for the range of blocks to be returned, and they will include the key facts in the query. This allows each storage cell to process the data request in the most efficient way possible and return only the data necessary to satisfy the query. It can filter out those columns from the data it ships back which reduces the total amount of data returned to the database compute node,
thus speeding up the IO operation. It is also capable of processing simple filters based on comparison operators contained in the WHERE clause and also many built-in single-row SQL Functions.

Exadata Smart also uses the Flash Cache feature of the Exadata Storage Server Software intelligently caches database objects in flash memory, replacing slow, mechanical I/O operations to disk with very rapid flash memory operations. The Exadata Smart Flash Cache holds frequently accessed data in very fast flash storage.

The Oracle optimiser has a new set of access paths that reflect the access to the storage cells. These access paths include some of the following:

- Table Access (Storage Full)
- Index (Storage Fast Full Scan)
- Table Access (Storage Full First Rows)
- Index (Storage Fast Full Scan First Rows)

All of these access paths use direct scans at the storage cell layer and uses parallel slaves to execute these scans.

In addition to this, Oracle has also introduced a number of new Oracle wait events such as

- Cell Smart Table Scan – Table Scan using smart scan on the storage cell
- Cell Smart Index Scan – Index Scan using smart scan on the storage cell
- Cell Multi Block Physical Read – Full Table Scan or Index Fast Full Scan not offloaded
- Cell Single Block Physical Read – Conventional Index Access

**Using Precise**

Precise provides the ability to see these Exadata access paths and wait events and identify what is consuming the most time.

Precise is a leading Application Performance Monitoring tool that provides a clear accurate breakdown of where is time spent in transactions. It identifies which tier is consuming the most time and allows investigation into that tier.

A key feature of Precise is its ability to identify where all the time is spent within the Oracle database and identify those SQL statements; database objects and Oracle wait events that are consuming large amounts of time.

This allows you to quickly identify where that time is spent and to focus on those SQL statements that are experiencing the worst performance.

Precise is able to show those key Exadata events that are consuming the most time and allows you to see how your database is performing on the Exadata machine.
Precise identifies these top events that are consuming the time as “Findings”. It tells you how much of the time is spent in this finding.

You can then investigate the finding further and identify the cause of the performance issue.

**Case Study**

In the following case study you can see that Precise has identified a number of findings. The finding of interest to us shows that 7% of the total amount of time is spent in RAC/OPS wait. This is indicating that we may have issues with the Exadata database.

![Case Study Image 1](image1)

By selecting this finding you can see that the time is spent in the wait event “cell single block physical read”. This is a new wait event specific to Exadata. You can see from the screenshot that this is the busiest wait event for this RAC instance and database.

![Case Study Image 2](image2)
The cell single block physical read wait event is a single block I/O.

We can also see which database objects are experiencing the high waits. In the example we can see that the I_OBJ2 index is experiencing the high waits.

Precise allows you to see which SQL statements are experiencing the high wait events as shown in the screenshot below. From this we can see that the statement 41340 is experiencing the high waits. We can also see that the number of executions is 1065 during this time frame and that the average response time is 1 minute and 8.5 seconds.

You can see the details for this SQL statement by selecting the TUNE icon. Precise then shows you the execution plan for the SQL statement. It shows very clearly the access path consuming the most time is spent. We can clearly see that the “Index (Storage Fast Full Scan)” on I_OBJ2 is contributing to 97% of the workload in the execution plan.
We can also see that there are “Findings” identified for this SQL statement. The Findings show the heavy time spent in the “Index (Storage Fast Full Scan)” as well as RAC wait and also “No Parallel Processes Available”.

Even though the execution plan is intending to use the Smart Scan access path, it appears that there are no parallel slaves available to carry out this scan of the storage cells. This could mean that the optimiser has to change the access path in order to read the blocks more efficiently.

This could therefore be the cause for high “Cell Single Block Physical Read”. This allows you to investigate the statement further and also check the parallel processes available.

You can also confirm whether the number of executions for this access path is too many and that the optimiser is being aggressive in choosing this access path.

Precise can also provide recommendations too for improving SQL queries as you can see in the screenshot below.

You can see there are a number of Exadata access paths in this execution plan with “Table Access (Storage Full)” and “Index (Storage Fast Full Scan)” being chosen by the optimiser.

You can see from the execution plan that the “Index (Range Scan)” on PS_PROJRESOURCE is contributing 96% of the time to it. This is the biggest contributor.

Precise is able to identify recommendations to improve this SQL statement as shown in the screenshot below. Precise has identified an index to improve this statement.
Summary

Exadata provides significant improvements in accessing the data in tables quickly for certain types of workload and is beneficial in many BI applications.

Precise helps you to identify those Exadata access paths and wait events. It helps you to decide whether these access paths improve the performance of the statements.

Precise also helps you to identify recommendations for improving these statements and provides you with clear information as to where that time is spent.