IDERA® | Precise Solution Brief

OBJECT OPTIMIZATION IN ORACLE WITH PRECISE

BY IDERA

OVERVIEW

One often addressed problems with application and database performance by upgrading hardware or database or even both. One could view that adding faster CPUs and more memory as an easy fix. Also, the database optimizer from each vendor has improved with each new release. But throwing hardware at a problem can only be a stopgap solution. And, we limit optimizers schema design, whether up-to-date statistics are available, and many other factors. Bigger and faster server purchases also resulted in larger data centers overloaded with servers, each with its own high cost and environmental impact.

With cloud computing, many companies have moved their resources to one of the different cloud providers to avoid these costs. This has made it possible to increase the capacity quicker and easier. However, if the virtual machine that the database runs on needs to be increased in size - CPU and memory, then this still comes at an additional cost as the size of the virtual machine (VM) increases.

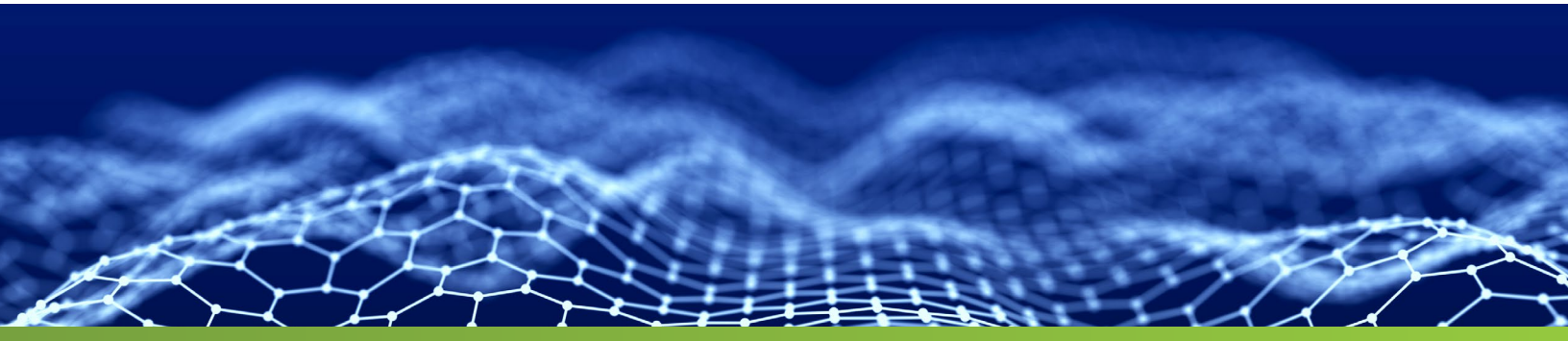
There is another way. We can reduce the need to increase the size of the virtual machine by optimizing performance. One did this by tuning the heaviest statements. Precise has a proven track record of providing alternatives to this approach. One of which is object optimization.

Many tuning methodologies tune one SQL statement at a time, adding indexes to cut down input/output (I/O) and speed up the query. However, each index has its own overhead. It consumes space and has to be maintained during update, delete, and insert operations. Each index also has to be maintained and sometimes rebuilt. To tune at the object level has the power to benefit all SQL statements accessing the most resource hungry tables. This can often deliver the required service-level agreement (SLA) performance much more quickly and at a much reduced cost. However, it is important to get the cost versus benefit analysis right.

The Account Receivable process in PeopleSoft shop can incur performance issues and has taken longer than 24 hours to complete. Precise spotted the bottleneck fast. They added sixteen indexes to the AR table. Index overhead can be tough to analyze alongside statement performance with conventional tools. But Precise can show the cost and benefit of each index, which indexes the optimizer used and which were not (that is, pure overhead). No other vendor pinpointed the root cause and fix it so fast.

This approach is similar for any database management system (DBMS) that Precise supports. These include Oracle Database, SQL Server, Sybase, and Db2. This document will talk about index optimization within Oracle Database.

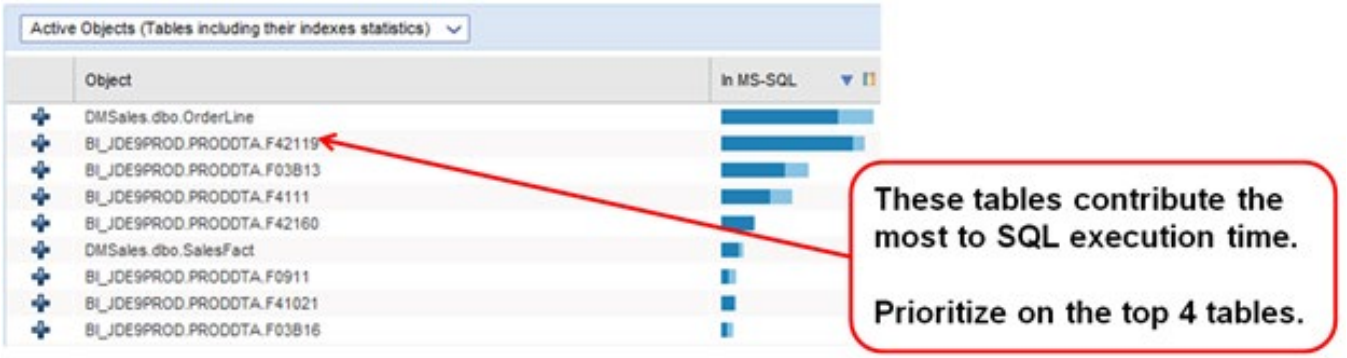
The formula is sufficiently simple: The ideal index configuration for an application is to place the minimum number of fresh, selective, low clustering factor indexes to minimize hardware contention.



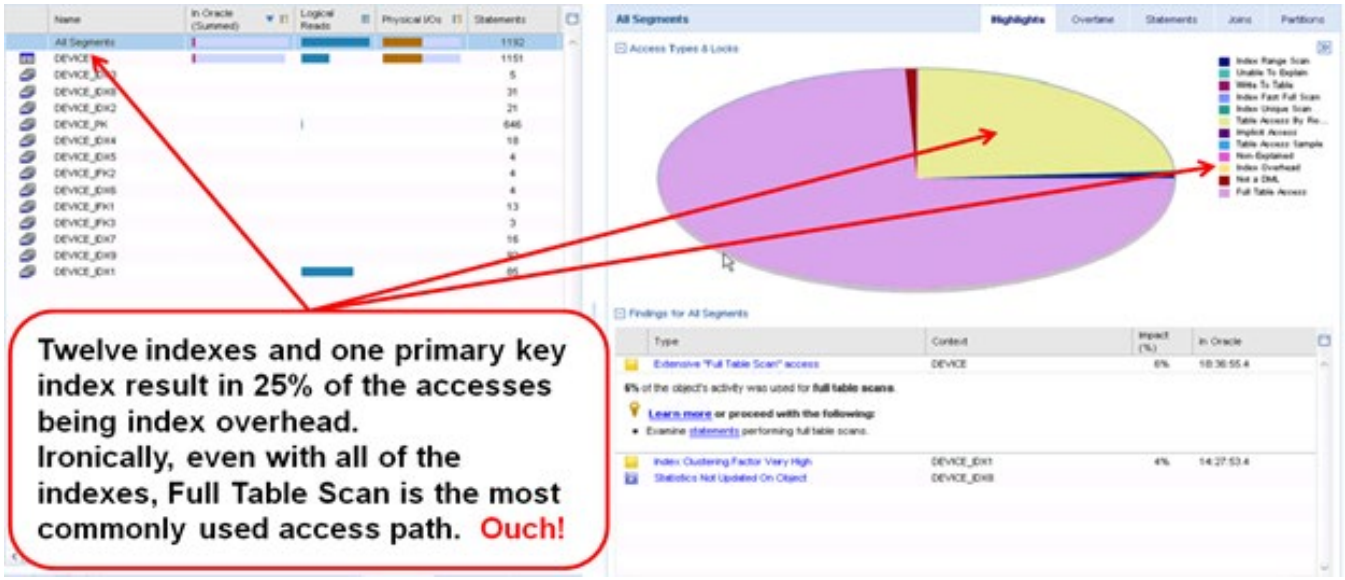
IDENTIFY THE CONTENDERS

The first step is to identify the tables which are contributing the most to overall SQL execution time. By using Precise, object tuning is a simple process. With accurate statistics gathered over time, Precise can identify the table access fast using the most resources and contributing the most to any poor application performance.

The example below shows some JD Edwards tables with the busiest tables prioritized at the top.



Once we have our candidate tables for tuning, we can check each one against each of the criteria we identified for optimum indexing.



The above example shows the table has twelve indexes and a primary key index. Yet with all this indexing, the most commonly used access path is still full table scan and 25% of the access is index overhead. The intelligent findings of Precise also show a very high clustering factor.

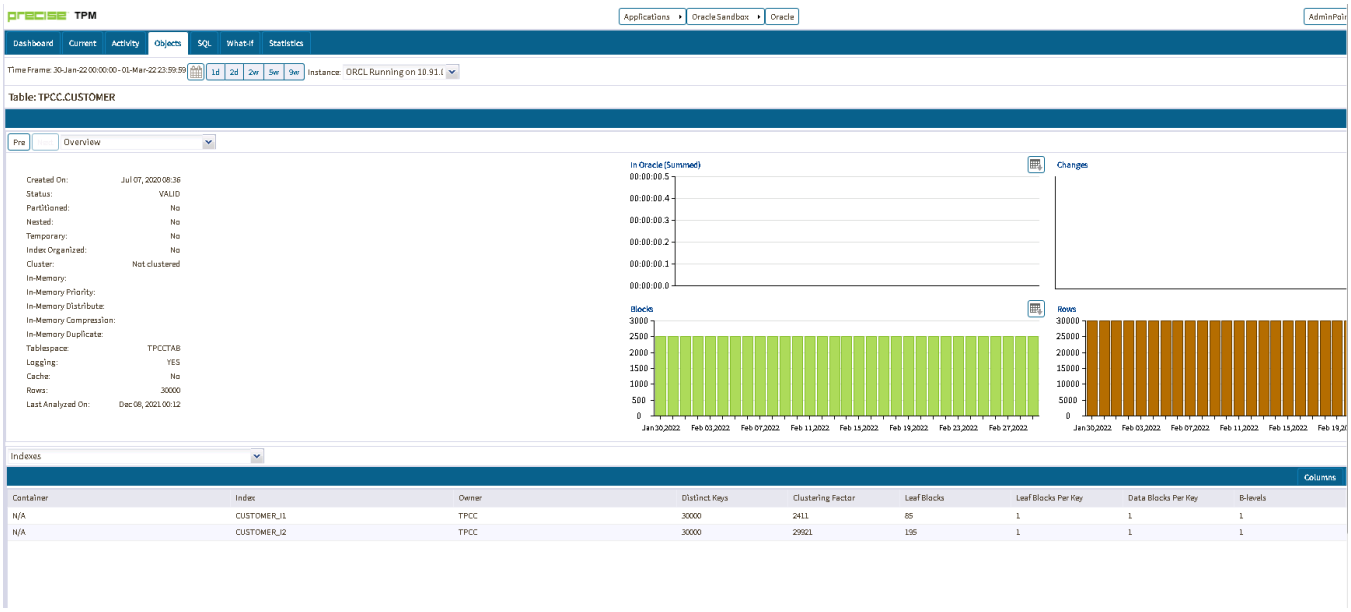
CLUSTERING FACTOR IS IMPORTANT

Clustering factor is a number representing the amount of order of rows in the table based on the values in the index. Where all the index entries in a leaf block point to the same block in the table, then the table is well ordered regarding the index, resulting in less input/output (I/O) operations using the index for access. This is a low clustering factor.

When all the index entries in a leaf block point to different blocks in the table, then the table is not well ordered, which will cause more I/O operations using this index for access. This is a high clustering factor.

If the value is near the number of blocks, then the table is well ordered. Here, the index entries in a single leaf block point to rows in the same data blocks. If the value is near the number of rows, then the table is arbitrarily ordered. Here, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.

Precise will show you the cluster factor for each index of the table.



By keeping the cluster factor for an index low, we can persuade the cost-based optimizer (CBO) to use this rather than a full table scan. However, Oracle can miscalculate the cluster factor, especially when inserting rows into a table using automatic space segment management (ASSM).

We may have to force the cluster factor to be low by using either TABLE_CACHED_BLOCKS statistic factor or attribute clustering.

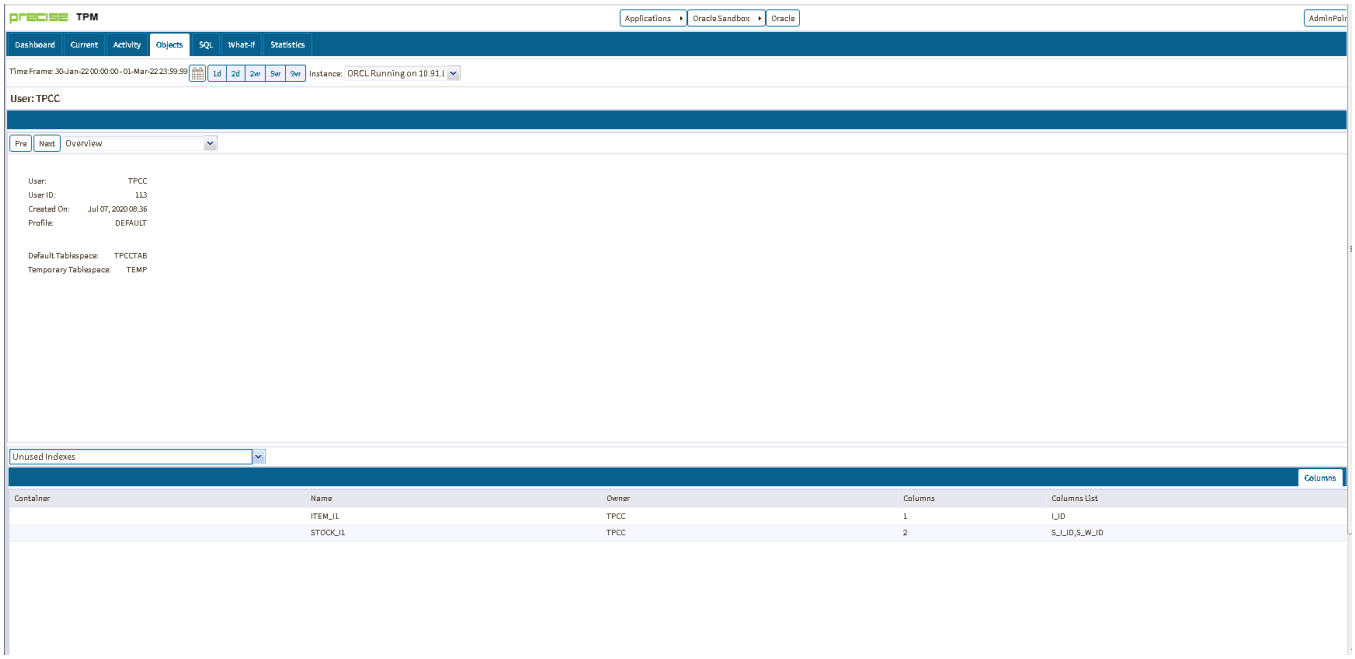
TABLE_CACHED_BLOCKS tells Oracle not to increment its block-change counter if the latest table block address matches one from the recent past. We recommend a maximum value of 16 for this.

ATTRIBUTE CLUSTERING is a table-level directive that clusters data in physical proximity based on what certain columns contain. Storing data that belongs together in physical proximity can reduce the amount of data to be processed and can lead to better performance of certain queries in the workload.

UNUSED INDEXES

Enterprise resource planning (ERP) solutions have many indexes created on tables. It is likely that application does not use many of these indices.

Precise provides a way to identify those unused indexes. It is then possible to look at removing those indexes and eliminate excessive index overhead. One should carry out testing before one does this in a production environment.



HAVE SELECTIVE INDEXES

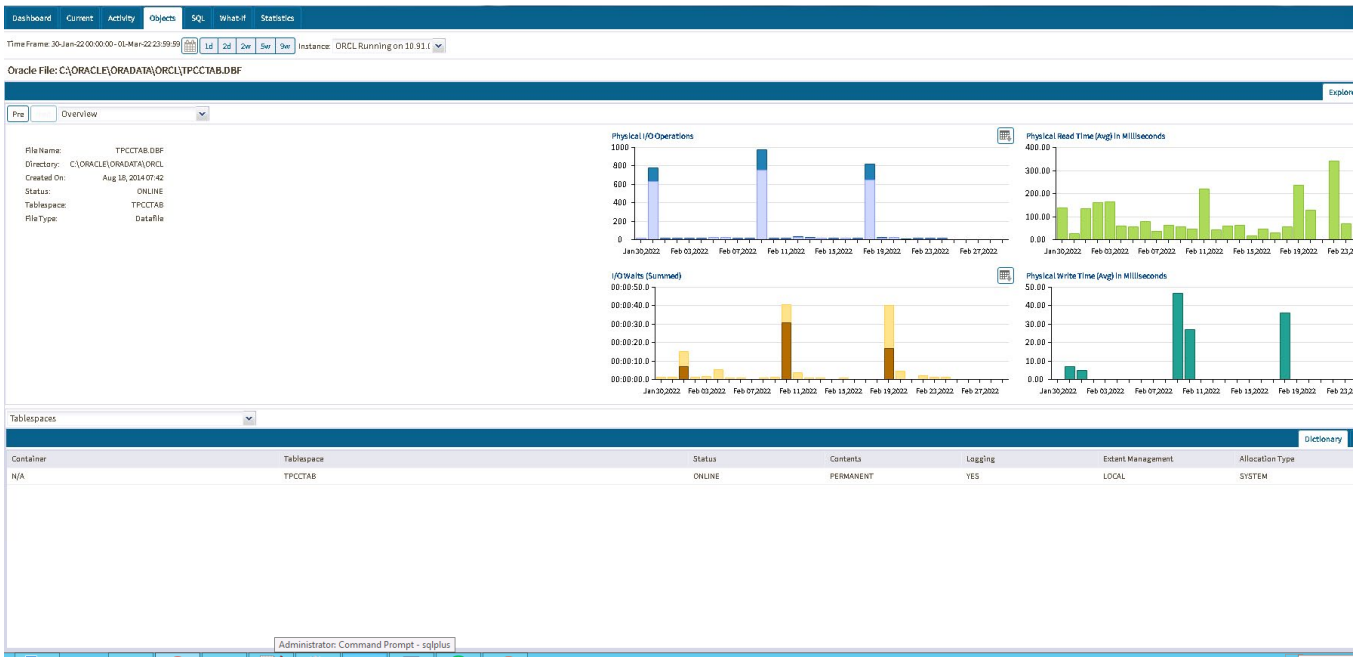
Executing SQL statements are working or waiting on a resource. Precise measures the contribution made by input/output (I/O) access to the object on the oracle data file and aggregates it into the category “I/O wait”. Where appropriate, this includes the entire storage area network (SAN) round trip time. In the example below, the database spends the majority, some 90%, of the execution time waiting on I/O.

The I/O wait time is the time spent accessing the oracle data file(s). This high I/O wait time could be because of several reasons:

- A large amount of I/O because the execution plan is poor and as a result, too many blocks qualify and need to be loaded into the Oracle buffer.
- Slow I/O, which can result from slow read I/O from the Oracle data file.

You can use the tune functionality of Precise to show the execution plan for the SQL query and identify possible recommendations.

Precise can also show I/O activity for the Oracle database files, which will help show if you have a storage problem with slow read I/O.



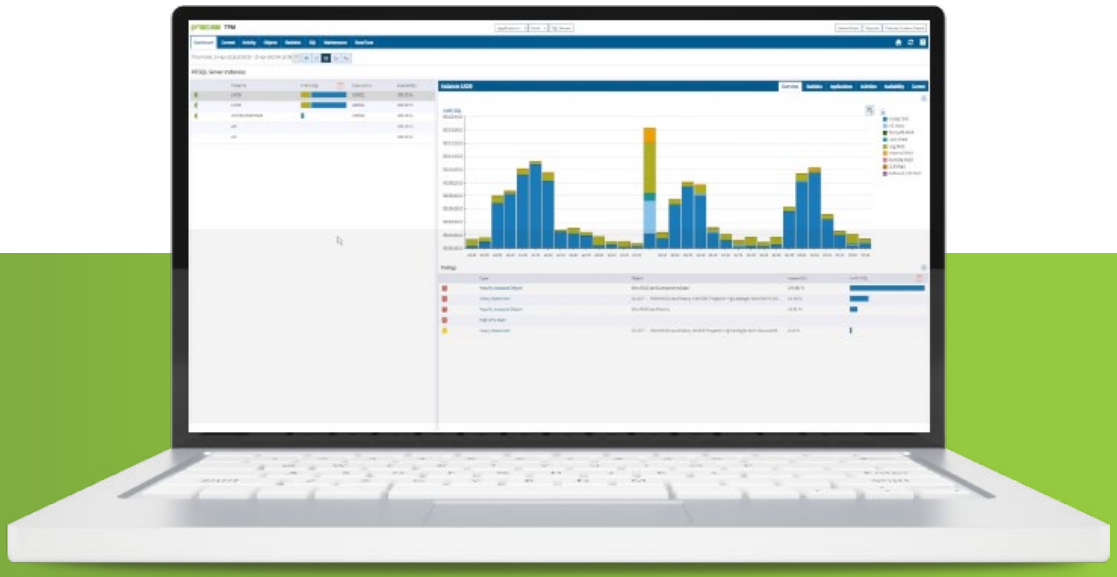
SUMMARY

Vendors can often deliver enterprise resource planning (ERP) applications with default indexing and insufficient maintenance routines to keep the execution plans and indexes healthy. Indexing can become out of date fast as applications develop and as users write their own reports and queries. To tune at the object level across the database as part of a proactive performance management policy is the best way to keep databases performing at their optimum. This also heads off any potential performance cliffs that are on the horizon. You can access the comprehensive and valuable data stored inside the Precise Performance Management Database (PMDB) via the graphical user interface (GUI) and simple to use dashboards, reports, and alerts. The PMDB speeds up this process and provides a valuable return on your investment in the quickest possible time.

PRECISE

Precise helps database and IT administrators to find and fix database and application performance problems in physical, virtual, and cloud environments. Unlike its competition, it provides deep database optimization, end-to-end transaction view, isolation of problems and causes, scalable deployment, what-if analysis for changes, and history, trending, and planning.

Start for Free



IDERA

IDERA.com