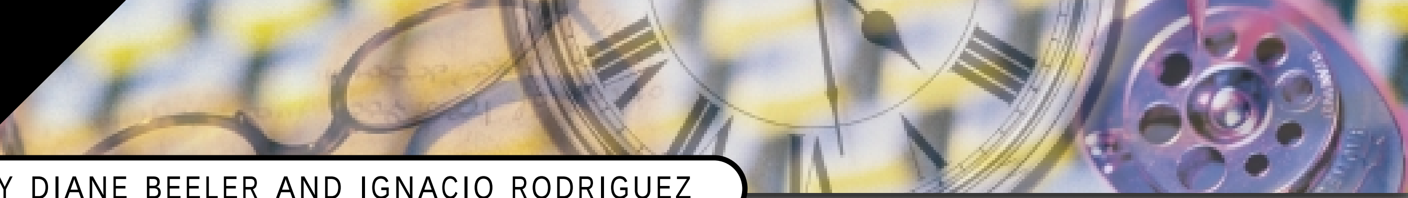
As appeared in
technical SUPPORT
Supporting Enterprise Networks and Operating Environments
December 2002



BY DIANE BEELER AND IGNACIO RODRIGUEZ

Database Performance Optimization Made Easy

Maintaining database performance and availability is a challenging task. Many events can affect database performance. Once these events occur, it is important to diagnose and resolve these issues quickly and easily. This article examines the steps a DBA can take to handle performance problems.

CUSTOMERS — and managers of the companies who serve them — demand instant, fast access to information. This means that the database administrators (DBAs) who are responsible for this information must optimize the performance and availability of databases and prevent bottlenecks. These DBAs must be familiar with multiple database platforms, operating systems and hardware platforms — with little or no training.

Maintaining database performance and availability is a challenging task. Many events can affect database performance. Once these events occur, it is important to diagnose and resolve these issues quickly and easily. Most importantly, it is essential to be proactive and notify the DBAs before these events actually occur and impact the users.

Let's examine how DBAs normally handle performance problems.

MANUAL OPTIMIZATION OF DATABASE PERFORMANCE

One of the most common reasons for poor database performance is un-tuned SQL. Many rules and guidelines need to be followed when queries are being written. If these rules are not followed, performance issues will occur. Some of these rules are as follows:

- For If statement(s) use the NOT IN operator, and rewrite them with NOT EXISTS.
- Compose Predicates Using AND and =.
- Use equijoins (a join with a join condition containing an equality operator).
- Avoid a full-table scan if it is more efficient to get the required rows through an index.
- Avoid using an index that fetches 10,000 rows from the driving table if you could instead use another index that fetches 100 rows and choose selective indexes.
- Choose the join order so you will join fewer rows to tables later in the join order.
- Set up the driving table to contain the filter condition that eliminates the highest percentage of the table.
- Use untransformed column values.
- Do not use SQL functions in predicate clauses or WHERE clauses. Use function-based indexes where possible.
- Avoid mixed-type expressions.
- When using IN and NOT IN with a subquery try to rewrite it with WHERE (NOT) EXISTS as an alternative.
- Minimize the use of DISTINCT.
- Do not use inequality operators such as '<>', '=!'.
- Do not use 'IS NULL' and 'IS NOT NULL'.

These rules can take the DBA anywhere from a couple of minutes to several hours each to analyze the SQL when it is done manually.

When an SQL performance issue occurs in production, the DBA or developer attempting to address the issue does the following:

- checks the SQL to make sure the rules listed previously are being followed
- possibly rewrites the query
- re-runs EXPLAIN PLAN to see if a better access path is selected

TYPICAL MANUAL TUNING STEPS

Here are the typical steps a DBA follows to tune SQL manually once he knows there are performance problems:

1. Locate the problem queries.
2. Run EXPLAIN PLAN.
3. Interpret the output. What is the operation order? Look at the full table scans. Look at join operations.
4. Look at the indexes. Are there too many? Are any of the indexes not being used? Is there no index?
5. Rewrite the query.
6. Run EXPLAIN PLAN.
7. Interpret the output. What is the operation order? Look at the full table scans. Look at join operations.
8. Compare with the previous EXPLAIN PLAN(s).
9. Is it better? If not, repeat all of the steps again.

This process can take a while, depending on the tuning experience of the DBA or developer. To get the best performance, he would have to manually rewrite a query several different ways, run the query, and then record statistics on how fast it ran. Did it improve? Did it improve enough? Was it fast enough? The DBA can spend many hours trying to tune the SQL, and in the end, discover that it was an application design issue, not a SQL problem.

There are issues that go beyond rewriting a query. EXPLAIN PLAN can show that the query is using the indexes that it is supposed to, but the performance is still bad. The potential cause could be unbalanced indexes, which can greatly impact query performance. If there are a large number of deletes on their associated tables, there will be unbalanced indexes. These indexes need to be rebuilt on a regular basis.

USING PROACTIVE AND DIAGNOSTIC TOOLS TO OPTIMIZE DATABASE PERFORMANCE

Some of the events that can influence database performance and availability are database configuration parameters, migrated rows, resource contention, and most important, un-tuned queries. Two types of tools are needed to ensure acceptable levels of performance and availability: proactive and diagnostic. The proactive tools alert DBAs to problems that are approaching unacceptable levels, and the diagnostic tools analyze and pinpoint problem areas and provide advice as to how to resolve issues.

With the right tools, DBAs can substantially increase the performance and availability of their databases. When purchasing tools, DBAs should look for those that have built-in intelligence and automation. These expert DBA tools can advise the DBA and make it easier for him to be effective and efficient.

Diagnostics and Optimization Tool

Database professionals of all experience levels need to manage and master the everyday challenges of database administration. The easiest way to do this is with a tool that provides database-at-a-glance diagnostics information on the health of your database. A Web-based tool like this that is accessible from any PC via your browser:

- improves the performance and prevents outages of a database by identifying and eliminating database bottlenecks
- delivers ready access from anywhere, anytime through a Web browser
- provides detailed drilldown information on user session details, 24-hour database view, 10 data files with highest I/O hits, four top memory consumers, space management, delays caused by redo log space requests, locks blocking other sessions, transactions that must wait on a rollback segment for completion, waits and latches, and many other critical metrics
- includes access, even remotely via the Web, to Oracle, DB2 UDB, and Microsoft SQL Server databases from a single console

Space Management Tool

Another weapon DBAs need in their arsenal is a tool that can analyze a database, predict problems, and help the DBA fix them before they even happen. This tool should provide information on when reorganization is needed and can even do the reorganization online. Additionally, this tool should:

- reorganize 24/7 databases online, without downtime, providing complete read and write access to database applications
- avoid unplanned outages by identifying objects that have high severity problems and provides recommended actions
- reduce the need for future reorganizations by performing powerful analysis of object condition, producing ideal settings that restore the object to peak performance
- perform automated sophisticated scheduling, object selection, multi-processing, and full checkpoint restart capability

Tuning Tool

DBAs need the most advanced SQL tuning technology on the market to improve business application performance by proactively analyzing, diagnosing, managing, and tuning databases and applications offline while preserving application and business availability. This tool lets DBAs collect data non-intrusively, analyze SQL statements, optimize a database, and do “what-if” analysis with exclusive simulation technology without affecting the production environment. This sophisticated tuning tool:

- prevents production interruption by using a non-intrusive high-speed data collector for capturing and saving SQL for later analysis
- extends DBA resources and improves application performance by automatically providing expert tuning recommendations
- reduces tuning time by displaying all relevant tuning information in an intuitive format
- tunes SQL before releasing application
- optimizes SQL that is tuned
- provides indexing suggestions based on SQL execution rebuilt

TYPICAL TUNING STEPS USING DBA TOOLS

When the DBA uses no tool, he typically does not know there are performance problems until after they have already occurred. With a diagnostic tool, he can instantly spot a performance problem on the database-at-a-glance screen. He can also be proactive and head off performance problems with the information provided in the monitors on the diagnostic tool screen. Once he figures out that he has performance problems — or that he will have performance problems if he does not take action now — he can use a tuning tool and a space management tool to optimize the performance of his database.

Here are the steps a DBA follows to tune SQL using tuning and space management tools.

1. Use a tuning tool which automatically and quickly:

   - identifies queries that are affecting the performance of the database

- gathers history via a High Speed Data Collector with very low overhead
- runs EXPLAIN PLAN
- displays several suggested query rewrites and a plain English explanation of execution steps
- provides table definitions and defines indexes
- uses simulation technology to run trials to compare execution statistics
- has no impact to production
- works great for long running queries
- identifies the best query to use
- compares EXPLAIN PLAN of the re-written SQL with the original SQL and color codes the differences

2. Use a space management tool to determine if indexes are selective or not and whether the indexes need to be rebuilt.

VENDOR APPLICATIONS

When using vendor applications such as SAP, Siebel, and Oracle Financials, the SQL in the application cannot be modified. However, you can achieve major performance gains by using tools that have the capacity to analyze index usage and create indexes based on specific usage patterns in your database. You can use these tools to identity poorly written SQL in these applications. Even though a DBA will not be able to change the SQL using many of these tools due to the restrictions in those applications, these tools are able to identify problem SQL, which allows the application vendors to fix it.

CONCLUSION

Often times, when a customer is experiencing a performance problem, the DBA's gut feeling is that there is something wrong with the database, even though they may not be able to identify the issue. A DBA can spend several hours a day trying to determine where the problem originated.

DBAs can ease their workloads and ensure that their databases are available and tweaked for top performance by using intelligent, automated and integrated DBA tools.

These tools make DBAs' jobs easier, dramatically reduce the time to solve and prevent performance problems, and ensure that their databases are available and operating at peak performance.

Diane Beeler has nearly 18 years experience in the IT industry. She has served as senior product manager, marketing manager, and marketing consultant for several major hardware and software companies. She currently serves as Consulting Product Marketing Manager in the Enterprise Data Management business unit of BMC Software.

Ignacio Rodriguez has more than 15 years of experience in the IT industry. He has served as an Oracle consultant for a major computer manufacturing company and for a variety of companies in the oil and health care industries. Ignacio most recently served as Senior Oracle DBA, Senior Software Consultant and Technical Product Manager for BMC Software. Currently, Ignacio is doing consulting in the Oracle world.

For subscription information,
email mbrship@naspa.com
or call 414-768-8000, Ext. 116.