

DB2 TUNING FOR PERFORMANCE



Jackie Jansen

This month we will investigate a few of the different tuning mechanisms available for improving the performance of queries and SQL statements.

Given a one page column all I can do is highlight areas and parameters that require more attention on your part. Query performance is largely determined by the choices the query optimizer makes. Your job is to ensure that the optimizer has the best information and help possible. In an earlier column you were introduced to the fact that, since V5R2, there have actually been two query optimizers in i5/OS and OS/400.

The newer optimizer, the SQL Query Engine or SQE is the preferred route for your queries. One of your tasks is to try to ensure that your queries can use SQE. This may mean moving to i5/OS V5R3 or V5R4.

Depending on your release and PTF level, different SQL syntax may prevent your query from using SQE. See the IBM website for more details. The Classic Query Engine or CQE is always used when you query logical files directly. In addition SQE does not support physical files that have logical files over multiple members or logical files that use select/omit logic. There is an option in the QAQQINI table where you can specify that the optimizer is to ignore these derived logical files when processing a query.

The most important, cost-effective step you can take to improve your overall query performance is to implement a good indexing strategy. Indexing strategies have been discussed in multiple columns over the last few years. As a minimum you want to create binary radix indexes over columns that are used in local selection (WHERE clause) or are join columns. Other good candidates are combining local selection and join columns with columns used for ordering and/or grouping.

Encoded Vector Indexes or EVIs can be particularly helpful when created over single local selection columns and join columns particularly in a star schema design. Non unique key columns with lots of duplicate values are also good contenders for EVIs.

In V5R4, for queries that use SQE, the index advisor has been greatly enhanced and can be of considerable assistance.

Starting in V5R3 with the appropriate database group PTFs IBM implemented Materialized Query Tables or MQTs. MQTs are also known as automatic summary tables. If an MQT exists the optimizer will evaluate using it in place of the much larger base table actually being queried. This can result in considerable time savings. MQT support is enabled via the QAQQINI file.




i5/OS has long supported the optional DB2 Symmetric Multiprocessing feature or SMP. If your company has an i5 550/570/590 or an iSeries 825/870/890 and the Enterprise Edition version of the software you may find that you already own this feature.

If you have a two-way or larger partition you should enable DB2 SMP. To do this use the QAQQINI file or the system value QQRDEGREE. If you want your job to take maximum advantage of all available processing power and memory use QQRDEGREE(*MAX). If you want your job to use parallelism but remain a "good neighbor" and not take more than its fair share of the resources specify QQRDEGREE(*OPTIMIZE). CQE and SQE calculate fair share somewhat differently. For CQE an often overlooked

important value in this fair share calculation is the MAX ACTIVE setting for your memory pool. If you allow this value to creep too high then you are restricting the amount of memory available to your query. Basically, for CQE, the system divides the pool size by the MAX ACTIVE value and calculates your job's fair share.

When using the Client Access ODBC driver there are quite a few parameters that you can use to help performance. These include Lazy Close, Block fetch, Query Optimization Goal (V5R4), and extended dynamic support. A good value to change, for read only applications, is the record blocking parameter. The default is 32kb. Consider increasing this to 512kb or even 1024kb.

There are many different tools to help analyze your query execution. Many of these were addressed in last month's DB2 Feedback Areas column.

For more information about DB2 Tuning for Performance go to <http://ibm.com/support/techdocs> and search on "Jackie Jansen". You will find a presentation originally given at the TEC2006 conference on just this subject. You will also find copies of previous TUG columns that address many of the individual recommendations in much more detail. 

Jackie Jansen is a Senior Consulting IT Specialist. She currently works in the IBM Americas Advanced Technical Support Solutions Centre. Jackie is a frequent speaker at iSeries Technical Conferences and User Group meetings. Contact her at jjansen@ca.ibm.com.