May 2013

zData Perspectives

General Guidelines for LOAD, REORG and REBUILD Utilities

by Craig S. Mullins

The LOAD, REORG and REBUILD utilities all move data around. As such, there are some general recommendations that apply to most of these utilities. Let's examine a few of them.

Accurate statistics are necessary to maintain current information about your data for access path determination. It's vital you update your statistics after loading or reorganizing significant amounts of data.

If you're loading data into a DB2 table specifying LOAD REPLACE, you should also consider the STATISTICS keyword to gather statistics as the LOAD utility executes. Because you're loading a table from scratch and any previous data will be lost, the statistics in the DB2 Catalog are obviously obsolete. Gathering statistics during the LOAD will outperform loading and then running RUNSTATS because you're reducing the number of passes against the same data.

help reduce the contention on this single latch. Taking an inline image copy in this case is recommended.

Compression and KEEPDICTIONARY. Running the REORG and LOAD REPLACE utility will rebuild the compression dictionary for table spaces defined with the COMPRESS YES parameter. Specifying the KEEPDICTIONARY parameter causes the utility to bypass rebuilding and reuse the existing compression dictionary.

This can improve the overall performance of the REORG utility because the CPU cycles used to build the dictionary can be avoided. However, if there's a significant change in data (e.g., loading completely different data) that could result in a significant change in compression dictionary, then the compression dictionary should be rebuilt to maintain a good compression ratio.

REPORTONLY. If the REPORTONLY keyword is specified for a REORG (table space or index), a report is generated,

same data.

On the other hand, if you're loading using RESUME YES, execute the RUNSTATS utility immediately after the LOAD completes.

Similar to generating inline statistics during a LOAD, you can also generate statistics during the execution of the REORG utility. By generating up-to-date statistics during the REORG instead of requiring an additional RUNSTATS step, you can improve performance. To generate inline statistics, simply specify the STATISTICS keyword on your REORG job. You can gather table space statistics, index statistics or both.

If you don't gather inline statistics or run a separate RUNSTATS job after reorganizing, the statistics in the DB2 catalog won't reflect the current state of the data and suboptimal access paths will likely be formulated by BIND and dynamic SQL.

Running inline statistics can reduce elapsed time anywhere from 0 to 35 percent compared to running a LOAD, REORG or REBUILD INDEX followed by a RUNSTATS.

To log or not to log. The LOAD and REORG utilities provide the option of turning logging on or off. If you specify LOG NO, DB2 won't log changes as the utility operates. In general, it usually makes little sense to enable logging during loads. Performance will be improved by avoiding the overhead associated with logging. Furthermore, the use of LOG NO should reduce the overall amount of log data generated. When there's a high log latch contention (latch class 19), the use of LOG NO can also

indicating whether or not the REORG should be performed. The actual REORG won't be performed. You can use the REORG utility in conjunction with REPORTONLY and the INDREFLIMIT and OFFPOSLIMIT keywords to produce reorganization reports.

Additional Guidance

For LOAD and REORG jobs that are I/O-bound, consider using Parallel Access Volume and BSAM/VSAM I/O striping.

In a data sharing environment, try to avoid running partition utility jobs in different members, if possible. Doing so will minimize the data sharing overhead in the Build phase, especially if there's a high index tree P-lock contention due to index splits. This can be seen in class 6 latch contention in the DB2 Statistics Report.

Keep in mind, too, that the maximum number of tasks for parallel LOAD or REORG is limited by the processor model, DBM1 virtual storage below 2 GB, number of partitions and number of indexes.

Effective plans for running your DB2 utilities in an efficient manner can have a significant positive impact on the performance and availability of your DB2 systems and applications. Remember, the impact is twofold: Running the right utilities at the right time will improve data availability and application performance; and running every utility using the proper parameters and options will reduce the overhead of utility execution.

Privacy Policy   Contact Us