

UPDATESTATS

PeopleSoft design introduced 'updatestats' facility during the PeopleTools Ver 8.1 release. Since then it has become a topic whether to enable updatestats or not. We will discuss the advantages and disadvantages here.

PeopleSoft batch processing

During batch processing, the programs usually squeeze out a subset of records from a fact table, insert into a temporary table, process it and then delete at the end of processing. This processing logic can repeat many times within the process. There has been a significant evolution in the DB2 SQL optimizer and it is a known fact that "the better the statistics are; the better the access path" when the program runs. Moreover, since the SQL is dynamic, an access path is very susceptible to changes in catalog statistics.

Queries involving two or more temporary tables, pose a challenging problem while tuning. Proper statistics either seeded or accurate are necessary to avoid a potential Cartesian product. Otherwise the response time will be too poor. GL Allocations, Summary Ledger and nVision queries rely on maintaining a proper join sequence.

A fairly commonly used technique to influence the statistics is to do one of the few things as described below.

- 1.0 Try RUNSTATS with zero records on a table. This almost certainly guarantees that the table will be chosen first in the join sequence because the optimizer expects zero rows from that table.
- 2.0 Try no RUNSTATS or default statistics. This is also referred to -1 statistics. Optimizer will use default values to determine what is best.

In both the cases above, what if you have two or more temporary tables to seed the statistics. In order to maintain a proper join sequence it becomes necessary to seed the statistics with proper values. Blind seeding sometimes may not really work due to the lack of depth in the seeded statistics. While it is very much possible to provide a manual depth of statistics, a most common practice is to perform a RUNSTATS while the process is running so that you can have a fairly accurate set of statistics. You will have to make sure that the seeded values are saved somewhere so you can re-apply them later. This has another potential issue.

90% of the temporary tables have Process Instance as a column. The Process Instance value is incremented for each process. So if you have old catalog statistics for Process Instance, in SYSCOLDIST and SYSCOLUMNS the optimizer will invariably choose the wrong set of tables to start the join process. Many times it involves a Cartesian product and the response will be too poor. It can also use inefficient join methods to the context.

In those cases, Updatestats will be very useful. You can insert into the temp table and as soon as the insert is completed, invoke RUNSTATS using the DSNUTILS stored procedure.

Advantages of Updatestats

Since you have the most recent statistics, you can expect the optimizer to determine the best possible access path.

No manual work is involved to gather statistics. All you have to do is just enable DBFLAGS in the App Engine or COBOL process at the right place.

Very effective in queries involving multiple temp tables where preserving proper join sequence is crucial.

In order to setup the Updatestats, you will have to follow the enhanced install process during the install time. Even though the install manual says so, this is not a requirement and you can enable the updatestats at anytime. If you did enable during the install time, it moves a bunch of temporary tables into its own tablespace, saving you a lot of time and effort.

Contrary to what many many people believe, enabling 'Updatestats' does not have a negative effect on the elapsed time of the process because a typical RUNSTATS on a small table is relatively fast.

Disadvantages of Updatestats

The main advantage itself is its biggest disadvantage. It has the potential to change the access path. What if the path changes to the worst possible path because of skewed statistics and DB2 optimizer got bad information from the statistics?

All temporary tables involved in Updatestats should be in its own tablespace.

Never configure Updatestats globally for all the temporary tables. That will result in an inefficient access path unless you have really found that collecting statistics helps DB2 optimizer to choose the best path.

Recommendation

Follow the enhanced path during install. Test the 'Updatestats' functionality so you can use when it is required. For queries involving multiple temporary table references, (when zero or -1 statistics fail to provide you with the best access path) always remember to make sure that you test and evaluate the access path by enabling the 'Updatestats'.