Statistics–How and When? Ask the Oracles!

Christian Antognini
Trivadis AG
Zürich, Switzerland

During the Hotsos Symposium 2007 held in Dallas last March, Jonathan Lewis, Wolfgang Breitling and I were invited to host a speakers' panel on the subject "Statistics: when, why, how". After about one hour of discussion it was clear that we had different opinions on the matter. Later on, Iggy Fernandez, editor of the Northern California Oracle Users Group (NoCOUG, www.nocoug.org) Journal, contacted us and Mogens Nørgaard, proposing we each write a short text answering the question "Statistics: How and When?". His excellent idea was to ask well-known Oracles (as he calls them) the very same question and then publish their answers in the NoCOUG Journal with a view to comparing different opinions about a very specific subject. Below you will find my text as it was published, along with the others, in the August issue of the NoCOUG Journal.

The package dbms_stats provides many features to manage object statistics. The question is how and when should we use them to achieve a successful configuration? It's difficult to answer this question. Probably no definitive answer exists. In other words there is no single method that can be implemented in all situations. Nevertheless, let me share my thoughts with you in this area.

The general rule, and probably the most important one, is that the query optimizer needs object statistics that de-scribe the data stored in the database. Therefore, when data changes, object statistics should change as well. As you may understand, I am an advocate of regularly gathering object statistics. Those who are opposed to this practice argue that if a database is running well, there is no need to re-gather object statistics. The problem with that approach is that more often than not, some of the object statistics are dependent on the actual data. For example one statistic that commonly changes is the low/high value of columns. True, there are not many of them that change in typical tables, but usually those that change are critical because they are used over and over again in the application. So in practice, I run into many more problems caused by object statistics that are not up-to-date than the opposite.

Obviously it makes no sense to gather object statistics on data that never changes. Only stale object statistics should be re-gathered. Therefore it is essential to take advantage of the feature that logs the number of modifications occurring to each table. In this way we re-gather object statistics only for those tables experiencing substantial modifications. By default a table is considered stale when more than 10% of the rows change. This is a good default value. As of Oracle 11g, this can be changed if necessary.

The frequency of the gathering is also a matter of opinion. I have seen everything from hourly to monthly or even less-frequent gatherings as being successful. It really depends on your data. Nevertheless, when the staleness of the tables is used as a basis for re-gathering object statistics, intervals between two gatherings that are too long lead to too many stale objects and, therefore, the time required for the gathering could be too long. For this reason I like to frequently schedule the gatherings to spread out the load and to keep a single gathering as short as possible. If your system has daily or weekly low-utilization periods, scheduling gatherings during those periods is usually a good thing. If your system is a true 24/7 system, it is usually better to use very frequent schedules (many times per day) to spread out the load as much as possible.

If for some good reasons object statistics should not be gathered on some tables, as of Oracle 10g you should lock them. In this way the job that regularly gathers object statistics will simply skip them. This is much better than completely deactivating the gathering job for the whole database. Unfortunately, up to Oracle 9i there is no feature that can be used simply to achieve the same goal.

If you have jobs that load or modify lots of data, you shouldn't wait for a scheduled gathering of object statistics. Simply make the gathering for the modified objects part of the job itself. In other words, if you know that something has substantially changed, trigger the gathering immediately.
As of Oracle 10g, you should take advantage as much as possible of the default gathering job. In order for this to meet your requirements, you should check-and if necessary change-the default configuration. Since a configuration at object level is only possible as of Oracle 11g, if you have particular requirements for some tables, you should schedule a job that processes them before the default job. In this way the default job will simply skip them. Locks might also be helpful to ensure that only a specific job is re-gathering object statistics on those critical tables.

If a gathering leads to inefficient execution plans, there are two things to do. The first one is to fix the problem by restoring the object statistics that were successfully in use before the gathering. The second one is to understand why inefficient execution plans are generated by the query optimizer with the new object statistics. For that purpose, at first you should check whether the newly gathered statistics correctly describe the data or not. For example, it is possible that sampling along with a new data distribution will lead to different histograms. If object statistics are not good, you know that the gathering itself-or possibly a parameter used for the gathering-is the problem. If object statistics are in fact good, there are two more possible causes. Either the query optimizer is not correctly configured or the query optimizer is making a mistake. Over the latter we have little control. On the other hand, we should be able to find a solution for the former. In any case, you should avoid thinking too hastily that gathering object statistics is inherently problematic and, as a result, stop gathering them regularly.

Since 1995, Christian Antognini has been focusing on understanding how the Oracle database engine works. He is currently working as a principal consultant and trainer at Trivadis AG (www.trivadis.com) in Zürich, Switzerland. If he is not helping one of his customers get the most out of Oracle, he is somewhere lecturing on optimization. Christian is the author of *Troubleshooting Oracle Performance* (Apress 2008).