TWENTY YEARS OF PERFORMANCE TUNING

by Chris Lawson

TIP #8 Understand Flashback Queries

Despite the appealing name, "flashback," a flashback query can run very slowly. On a large production system, a flashback query going back a few hours can easily take 10 hours. This happens because Oracle must reconstruct an object as it existed at a certain time. This reconstruction happens one block at a time, going backwards in time, undoing each transaction.

Before Oracle can reconstruct an object, it has to *identify* what needs to be undone. One would think this is a trivial step, but not so. This can be very time-consuming--especially when the database has undergone lots of recent transactions.

Transaction Table

In each undo segment header there lies a critical structure known as the *transaction table.* It's not a "table" as we normally think of it. Maybe a "list" would have been a better name. The transaction table identifies the undo information held in that undo segment. For example, any given entry points to where to find the actual undo block.

That sounds excellent, but the entire transaction table only has information for 34 transactions. Each entry is called a transaction *slot*. As more transactions are housed in a given undo segment, transaction slots, being so few, are very often *overwritten*. The information is not lost, of course, but to find it, several extra steps will be required. On a very busy system, it could take thousands of extra reads just to find where to start.

Remember--all this effort is before Oracle even starts the "real work" of rebuilding the object of interest to the time desired. Of course, that final step will add even more time. The point is, the delay of determining where to start can be vastly more than the work required to actually *do* the reconstructing of the object.

When a transaction table slot is reused, what happens to the valuable information that used to be kept in that slot? Oracle stores the old slot information right at the beginning of the new undo block that used that slot. In this way, the information is linked together. Therefore, when we perform a flashback query, we can discover what the transaction table looked like at some prior state. Oracle calls this, "Transaction Table Rollback." You can also get a summary in the AWR report, in the *Instance Activity* section.

What can I do?

The essence of the problem is having to repeatedly reconstruct the contents of the "slots" in the transaction table. If there were fewer re-uses of the slots, then there would be less work required. Oracle support has suggested keeping more undo segments online--and therefore more slots available.

This is accomplished by setting the underscore parameter, ***_rollback_segment_count***. The idea is, to override the automatic undo process, and force more undo segments to stay online. It seems like the number of reused "slots" should go down commensurate with the extra undo segments that are kept online. So, if we keep 4x as many undo segments online, I would expect to see approximately a 4x reduction in transaction table rollbacks. That's the theory, anyway, but I haven't confirmed that yet.

Chris Lawson is the author of *The Art & Science of Oracle Performance Tuning*, as well as *Snappy Interviews: 100 Questions to Ask Oracle DBAs*. When he's not solving performance problems, Chris is an avid hiker, runner, chorister, Amazon reviewer, and geocacher. Chris writes using the penname, "*Bassocantor*."

Twenty Years of Performance Tuning is a series of tips based on the author's experience solving performance problems over the last 20 years.