

Is there a difference between commit and rollback in a transaction only having selects?

Asked 16 years, 2 months ago Modified 11 months ago

Viewed 11k times



21



The in-house application framework we use at my company makes it necessary to put every SQL query into transactions, even though if I know that none of the commands will make changes in the database. At the end of the session, before closing the connection, I commit the transaction to close it properly. I wonder if there were any particular difference if I rolled it back, especially in terms of speed.

Please note that I am using Oracle, but I guess other databases have similar behaviour. Also, I can't do anything about the requirement to begin the transaction, that part of the codebase is out of my hands.

sql

oracle-database

transactions

commit

rollback

Share

Improve this question

Follow

asked Oct 13, 2008 at 16:04



Tamas Czinege

121k ● 40 ● 153 ● 177



14



Databases often preserve either a before-image journal (what it was before the transaction) or an after-image journal (what it will be when the transaction completes.) If it keeps a before-image, that has to be restored on a rollback. If it keeps an after-image, that has to replace data in the event of a commit.

Oracle has both a journal and rollback space. The transaction journal accumulates blocks which are later written by DB writers. Since these are asynchronous, almost nothing DB writer related has any impact on your transaction (if the queue fills up, then you might have to wait.)

Even for a query-only transaction, I'd be willing to bet that there's some little bit of transactional record-keeping in Oracle's rollback areas. I suspect that a rollback requires some work on Oracle's part before it determines there's nothing to actually roll back. And I think this is synchronous with your transaction. You can't really release any locks until the rollback is completed. [Yes, I know you aren't using any in your transaction, but the locking issue is why I think a rollback has to be fully released then all the locks can be released, then your rollback is finished.]

On the other hand, the commit is more-or-less the expected outcome, and I suspect that discarding the rollback area might be slightly faster. You created no

transaction entries, so the db writer will never even wake up to check and discover that there was nothing to do.

I also expect that while commit may be faster, the differences will be minor. So minor, that you might not be able to even measure them in a side-by-side comparison.

Share Improve this answer

edited Jan 9 at 7:27

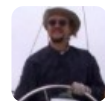
Follow



Pang

10.1k ● 146 ● 85 ● 124

answered Oct 13, 2008 at 16:46



S.Lott

391k ● 82 ● 517 ● 788

-
- 5 I don't recognise this as a description of the way oracle works at all. It sounds like a generic description that has been applied to Oracle. Guesses at how Oracle works are not likely to be helpful. – [David Aldridge](#) Oct 13, 2008 at 22:16
-

Oracle calls the journal a "Redo Log File". It calls the rollback segments an "Undo Tablespace". Do you know which is quicker? Commit or rollback? – [S.Lott](#) Oct 13, 2008 at 22:26

If there's no work to do then there's almost certainly no difference. Oracle is optimised to commit fast -- it only requires that the commit record be written to the redo log buffer and the buffer flushed (except for asynchronous commit in 10g+). A rollback is more work. – [David Aldridge](#) Oct 15, 2008 at 16:46

- 1 At asktom.oracle.com/pls/asktom/..., Tom Kyte indicates that the database will not actually do any commit work if there is no transaction. You could probably check that will by doing a few hundreds commits really quick and checking the count of log file sync events. – [Gary Myers](#) Sep 17, 2009 at 8:00
-



8



I agree with the previous answers that there's no difference between COMMIT and ROLLBACK in this case. There might be a negligible difference in the CPU time needed to determine that there's nothing to COMMIT versus the CPU time needed to determine that there's nothing to ROLLBACK. But, if it's a negligible difference, we can safely forget about it.

However, it's worth pointing out that there's a difference between a session that does a bunch of queries in the context of a single transaction and a session that does the same queries in the context of a series of transactions.

If a client starts a transaction, performs a query, performs a COMMIT or ROLLBACK, then starts a second transaction and performs a second query, there's no guarantee that the second query will observe the same database state as the first. Sometimes, maintaining a single consistent view of the data is of the essence. Sometimes, getting a more current view of the data is of the essence. It depends on what you are doing.

I know, I know, the OP didn't ask this question. But some readers may be asking it in the back of their minds.

Share Improve this answer

Follow

edited Jan 9 at 7:24



Pang

10.1k ● 146 ● 85 ● 124

answered Oct 13, 2008 at 16:43



Walter Mitty

18.9k ● 2 ● 31 ● 59



3

In general a COMMIT is much faster than a ROLLBACK, but in the case where you have done nothing they are effectively the same.



Share Improve this answer

answered Oct 13, 2008 at 16:15

Follow



Tony Andrews

132k ● 23 ● 227 ● 264



3

The documentation states that:



- Oracle recommends that you explicitly end every transaction in your application programs with a COMMIT or ROLLBACK statement, including the last transaction, before disconnecting from Oracle Database. If you do not explicitly commit the transaction and the program terminates abnormally, then the last uncommitted transaction is automatically rolled back. A normal exit from most Oracle utilities and tools causes the current transaction to be committed. A normal exit from an Oracle precompiler program does not commit the transaction and relies on Oracle Database to roll back the current transaction.

http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/statements_4010.htm#SQLRF01110

If you want to choose to do one or the other then you might as well do the one that is the same as doing nothing, and just commit it.

Share Improve this answer

edited Oct 14, 2008 at 17:50

Follow

answered Oct 13, 2008 at 22:18



David Aldridge

52.3k ● 8 ● 72 ● 99

It depends on the client actually. sqlplus, its an implicit commit. others it might not be. If the network connection is severed (eg, the client just "goes away"), then its a rollback.
– [Matthew Watson](#) Oct 14, 2008 at 9:14

I'm not sure that it does depend on the application, I should probably have said a "graceful disconnect", but the docs say that "An implicit request occurs after normal termination of an application or ..."

download.oracle.com/docs/cd/B28359_01/server.111/b28318/... – [David Aldridge](#) Oct 14, 2008 at 17:50

Actually, I found a better reference and edited my posting.
Thanks. – [David Aldridge](#) Oct 14, 2008 at 17:54



1

Well, we must take into account what an SELECT returns in Oracle. There are two modes. By default an SELECT returns data as that data looked in the very moment the SELECT statement started executing (this is default



behavior in READ COMMITTED isolation mode, the default transactional mode). So if an UPDATE/INSERT was executed after SELECT was issued that won't be visible in result set.

This can be a problem if you need to compare two result sets (for example debta and credit sides of an general ledger app). For that we have a second mode. In that mode SELECT returns data as it looked at the moment the current transaction began (default behavior in READ ONLY and SERIALIZABLE isolation levels).

So, at least sometimes it is necessary to execute SELECTs in transaction.

Share Improve this answer

edited Oct 21, 2008 at 20:46

Follow

answered Oct 21, 2008 at 20:27



pero

4,259 ● 29 ● 27



0

Since you've not done any DML, I suspect there'd be no difference between a COMMIT and ROLLBACK in Oracle. Either way there's nothing to do.



Share Improve this answer

answered Oct 13, 2008 at 16:07

Follow



[cagcowboy](#)

30.8k ● 11 ● 74 ● 94



0

I'd think a Commit would be more efficient; since generally you'd expect most DB transactions to be committed; so you would think the DB optimizes for this case (as opposed to trying to be more efficient for a rollback).



Share Improve this answer

answered Apr 7, 2011 at 10:52

Follow



[Andrew](#)

1,616 ● 1 ● 12 ● 19

