Internals of online index build

Saibabu Devabhaktuni

About me

- Sr Manager of DB engineering at PayPal
- Using Oracle since 1998
- http://sai-oracle.blogspot.com
- Can be contacted at saibabu_d@yahoo.com
- Lives in Fremont, CA

Scope

- Only B-tree index is covered
- No domain indexes
- Tested on Oracle 10.2 and 11.2
- Some of the observations here can be incorrect or may change in future versions.

Introduction

- Online index build (OIB) introduced in 8i
- Base table DML's can continue during OIB
- Parallel option is supported
- Enhanced in 10g
- Rewritten in 11g

Relevant index facts

- All indexes are unique (rowid implicitly added to key columns for uniqueness)
- Updates are converted to Delete and Insert in index
- Oracle can walk through across at Branch/Leaf block level

Relevant index facts (continued)

- Leaf block should have enough room to fit at least 2 records
- Leaf block splits carry forward ITL slots (may change in future, filed Oracle ER 8767925)
- Deletes doesn’t cause empty leaf blocks to unlink from index structure (they are only added to the freelist)

10,000 ft OIB view

- OIB process briefly lock base table
- Create journal table to track changes
- Let base table DML's continue
- Build index
- Merge journal table changes
- Lock the table again to finish final merge
- Drop journal table

OIB in Oracle 8i

- Rebuilding existing index online does FTS
- Offline index rebuild does index scan (NO FTS)
- OIB session does FTS as of OIB start SCN
- Possibility of ORA-1555 for big tables

OIB in Oracle 10g

- Both offline index build and OIB does FTS
- OIB session does FTS as of current SCN (similar to how consistent reads are done for DML’s) when the table blocks are read (demo)
- Very low possibility of ORA-1555
- Only one OIB at a time on a given base table
- Automatic stats gathering at build time

Monitoring OIB

- Query dba_objects for journal table (SYS_JOURNAL_object-id-of-new-index)
- Select queries allowed on journal table
- One record for each OIB in sys.ind_online$
- SMON cleanup any aborted OIB's in ind_online$
- Query v$session_longops for tracking OIB progress

Journal table

- Journal table is an IOT table
- Index key columns named as C0, C1,..etc.
- Rowid (RID) of base table added to IOT
- OPCODE column in IOT track inserts and deletes
- PARTNO column in IOT track base table partition
- Primary key consists of (C0, C1, .., RID)

Journal table (continued)

- Key columns of CHAR type converted to VARCHAR2 in IOT
- Unique index allow duplicates during OIB, error only at the merge phase (demo)
- Null key values tracked in OIB as not null values (demo)
- OIB may not work for larger key lengths, but offline index can work (demo)

Journal table (continued)

- Journal table is not partitioned
- Reverse or hash partition OIB doesn't create corresponding type of journal IOT (demo)
- Delete on base table doesn't cause same record to be deleted in journal table
- All DML's for a given record (by rowid and opcode) will only have final state recorded in journal table (demo)

Locking in 10g

- OIB starts off with joining lock queue (table lock request 4 (share), lock mode 2 (row share)) and create journal table with share lock on it
- Existing open transactions can continue
- New DML transactions will have to wait
- Once the open transactions commit, then OIB session removes type 4 share lock (row share lock remains)

Locking in 10g (continued)

- New transactions can continue
- OIB session place type 4 share lock on base table at the end of 1st merge phase
- OIB session wait for open transactions to end
- New transactions will wait during 2nd merge
- OIB session complete the 2nd merge, drop journal table and release all table level locks (demo)

Locking drawbacks in 10g

- Open transactions during OIB initialization will cause new transactions to wait
- Open transactions during OIB final merge phase will cause new transactions to wait
- Long running transactions during OIB can cause big impact
- Only one index can be built at a time per table

Merge in 10g

- At the end of index build, OIB session will start merging changes from journal table
- DML's on base table will continue to be tracked in the journal table
- OIB session will start reading the journal IOT table from left most leaf block
- OIB will walk across leaf blocks in journal table and mark them as deleted (1st merge)

Merge in 10g (continued)

- Once OIB reaches the last leaf block of journal table, new transactions will hang, but the existing transactions continue
- OIB wait for open transactions to end
- OIB session does 2nd phase merge of all journal table leaf block contents, but don't mark them as deleted
- Drop the journal table and release all locks (demo)

Merge drawbacks in 10g

- Very long merge time since transactions continue to write to journal table even after index build phase
- Long 2nd merge time possible with large open transactions
- Aborted OIB can leave journal table behind and new transactions still tracked in journal table.

Locking in 11g

- OIB starts off with joining lock queue (type 2 row share table lock only, no type 4 lock) and create journal table with type 4 share lock on it
- Existing open transactions can continue
- New DML transactions doesn’t wait at all
- Once the open transactions commit, then OIB start base table scan for building the index

Locking in 11g (continued)

- OIB session continue to keep row share lock on base table till the end of merge phase
- At the end of merge phase, OIB session wait for open transactions to end
- New transactions can continue uninterrupted
- OIB session complete the merge, drop journal table and release all table level locks (demo)

Locking benefits in 11g

- Transactions will never wait for OIB
- More than one index can be built at a time per table
- Long running transactions will only cause OIB completion to take longer
- It is not same as DDL_LOCK_TIMEOUT (in fact this was introduced in 8i for OIB only)

Merge in 11g

- At the end of index build, OIB session will start merging committed changes from journal table
- DML's on base table will continue and they go to target index (not to journal table)
- Ongoing DML changes also go to journal table if the base record already exist there
- In the merge phase, OIB session will start reading the journal IOT table from left most leaf block

Merge in 11g (continued)

- OIB will walk across all leaf blocks in journal table and mark records as deleted as they are merged
- Once OIB reaches the last leaf block of journal table, 1st mere complete, wait for open transactions to end
- Complete the 2nd merge after open transactions, started before end of 1st merge*, end (demo).

Merge benefits in 11g

- Faster index build due to transactions directly going to target index during merge phase
- Long running transactions have less impact on merge operation.
- 2 phase merge is more effective than on 10g, i.e. no transactions wait on OIB.
- Less prone for index contention (more on this later)

OIB drawbacks in 11g

- Aborted OIB require exclusive table lock for SMON cleanup
- DDL_LOCK_TIMEOUT doesn't work for OIB abort cleanup (bug 10038517)
- Hash partitioned (or reverse key) index to resolve contention will suffer contention during OIB since journal table can't be partitioned or reversed (Opened ER: 9912950)

More on OIB

- "_enable_online_index_without_s_locking" parameter for 10g/11g behavior
- DBMS_REPAIR.ONLINE_INDEX_CLEAN can be used for OIB cleanup, need exclusive table lock (demo)
- Event 10622 for tracing OIB
- Event 10626 for OIB timeout on DML

OIB Summary

- Suffers from locking and long merge times in 10g. Not ideal for some OLTP db's.
- Rewritten in 11g for better locking, faster merge, and faster index build
- Can be used on busy OLTP 11g db's also
- 11g still suffers from cleanup of aborted OIB
- Journal table is prone for contention during OIB