Consistent scanners

sershe, 2013-12-26

Scanners currently use mvcc for consistency. However, mvcc is lost on server restart, or even a region move. To enable the scanners to be consistent, we need to transfer mvcc between servers.

Note on numbers

HBase currently has 3 version numbers for each KV - sequence Id, mvcc, and timestamp. Timestamp is generally not used for consistency except by client app logic. MVCC and seqId should be redundant for all practical purposes, and may be merged (see HBASE-8763).
This document uses "**mvcc**" to mean either "mvcc" as it stands now, or a merged mvcc-seqId number.

Client changes

Client will have a configurable setting to enable consistent scanners. If this setting is enabled, mvcc will be returned with the scan response, and client will track that mvcc.

WAL changes

First of all, mvcc from memstore should survive during recovery. Thus, in addition to seqId, WAL will store mvcc for each edit. mvcc is obtained before WAL write so it's ok to store both. WAL writes do not have to be in mvcc order (unless mvcc and seqId are merged). Recovery will have to make use of mvcc when replaying the edit to new server.

Server changes

The main problem would be that the “lowest scanner” mvcc information may no longer be present at RS after it opens the region, because it may be on the client. To be able to maintain lower boundary for client-side mvcc values without adding a central authority/single point of failure, the grace period will be used after the region is opened. During this grace period, mvcc will not be discarded from KVs. Grace period will be based on scanner timeout; it is assumed that all interested scanners will "check in" with RS within this period and help it establish the real low boundary.
Given that grace period is not bulletproof, RS should be able to determine the lowest *allowed* scanner mvcc, given existing store files. The consistent scanners that request the mvcc below that would fail.

Storefile/compaction changes

The usage of existing "include mvcc" functionality for KVs will be extended to compactions in such manner that mvcc are not discarded from store files if some client-side scanners might be active (see above). As usual, mvcc-s above the current lowest scanner read point will not be discarded, and during grace period no mvcc will be discarded. Thus, if we have some mvcc present in store file, it means all higher mvccs are also present.

Thus, each store file will contain the lowest mvcc in the file (might be unset, which means all mvcc in the file are discarded) in the header. Using these, RS will be able to determine the lowest allowed scanner mvcc (see above).