# Bulk Load Replication in HBASE

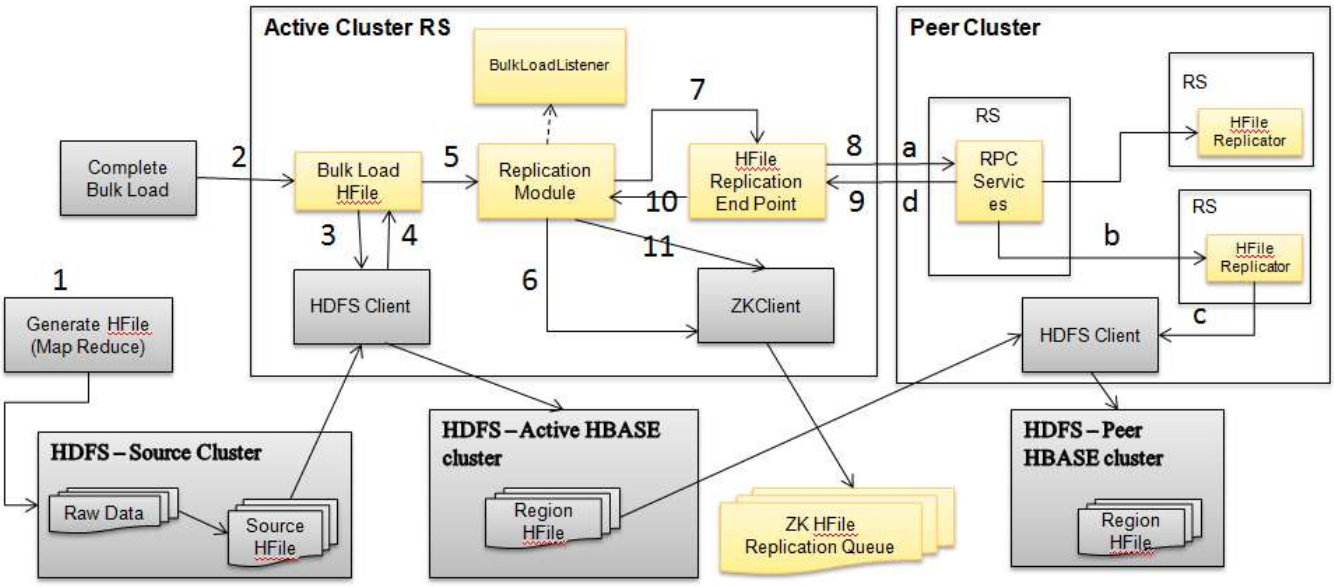
Ashish Singhi & Bhupendra Kumar Jain
Huawei Technologies
August 2015
v0.1

## 1 Requirement Scenario

Hbase Replication currently doesn't replicate the bulk loaded hfile data.

Currently replication happens thru replaying the WAL entries from Active cluster to peer cluster. Since Bulk Load process by-passes the WAL edits, so bulk load data will not be replicated.

## 2 High Level Solution




- HDFS- Source cluster: Contains RAW data and generated hfiles
- HDFS- Active HBASE cluster: HDFS cluster used by active HBASE cluster
- HDFS- Peer HBASE cluster: HDFS cluster used by Peer HBASE cluster

## 2.1 Solution

Replication module will be enhanced to support bulk loaded hfiles also. The bulk loaded hfiles path will be queued in ZK. Active Hbase RS will send queued hfile path to peer cluster to replicate. Peer cluster will load this hfile into its appropriate table Regions. [Similar to complete bulk load mechanism]

## 2.2 Detailed Flow

- Generating hfiles from Raw data thru Map-Reduce task (ImportTsv or others)
- Generated hfiles will be loaded into their proper table regions by complete bulk load.
- On successful bulk load of hfiles, RS will notify to all the registered BulkLoadActionsListener.
- Replication module will be one of the BulkLoadActionsListener, so it will get notification about newly added hfiles along with their hdfs paths.
- Replication Module will add this hfile entry into ZK queue and will invoke HFileReplicationEndPoint for replicating this new hfiles.
- HFileReplicationEndPoint will maintain a queue of hfiles. After every configurable interval or max request size limit, it will send a RPC request to peer cluster RS with all queued entries.
- Peer cluster RS will receive the RPC request having multiple hfile paths. It will distribute the hfiles to appropriate RS based on the table Region Splits in Peer cluster.
- If source hfile doesn't fit into any of the Region boundaries in peer cluster, then RSRpcServices will split the source hfile as per peer cluster table region boundaries. Here the split files will be written to "hbase temp folder in Peer HDFS cluster"
- Peer RS will send the response with Success OR Failure paths.
- Replication module will clean up the ZK queue according to the response.

**Points**

1) If compaction, Merge or split of hfile happens during the replication process in active cluster, then the older hfile (bulk loaded hfiles) will be moved to archive folder and replication module will refer to archive folder entry. The hfile should not get deleted from

archive folder until the replication is finished.

## 2.3 ZK Replication Queue Structure


- Replication queue for hfiles and wals will be maintained separately in ZK under /hbase/replication node for each RS.
- Inside hfiles node, there will be children node for every bulk loaded hfile name and hfile path as its data.
- Separate node will be maintained for wals. The structure inside wals node will be same as existing.

## 2.4 Failover Handling

a) Active RS goes down:

If one of the active RS goes down, the replication queue will be handled by another RS of same cluster. [Same as existing design]

b) Active cluster goes down:

During startup, each RS will rebuild the replication queue from ZK and will process

this queue separately. [Same as existing design]

c) Peer cluster RS goes down during replication:

ReplicationEndPoint will retry the replication to another RS of peer cluster. [Same as existing design]

d) Peer cluster goes down / Disable replication:

The new hfiles will be queued till its max limit in ZK. Once the limit is reached, the new entries will not be queued.

e) Active cluster RS goes down before receiving the Response:

Since ZK still contains these hfile entries, so the replication request will be sent again by another RS of active cluster.

# 3 Constraints and Limitations

- The max number of hfile replication nodes in ZK will be same as per ZK limits. Once limit reaches, new hfiles will not be queued.
- If data in Visibility Labels table is different in active and peer cluster, then Visibility expressions will fail during scan in peer cluster.

  Example:

  Active cluster visibility table has entry such as: SECRET 1

  Peer cluster visibility table has entry such as: SECRET 2

  Bulk loaded Hfile of Active cluster will contain the value as "1", this hfile will be directly copied to Peer cluster during replication, So peer cluster hfile will still have entry as "1". During Scan there will not be any matching entry corresponding to "1" in Peer cluster Visibility Tables. So scan will fail to parse this visibility expression.
- Cyclic replication: There will not be any data validation for cyclic case. So any hfile is replicated to cluster-2 and if again replicated from cluster-2 to active cluster, it will be accepted.
- Peer cluster will require the Read permission for active HDFS cluster.
- Source and Peer cluster must have same Compression codec for replicated tables.

# 4 Performance Scenarios

- RPC Calls: To reduce the Number of RPC calls, the replication request will be sent in parallel for multiple hfiles together.
- Hfile Data Transfer: To reduce the amount of data sent over the network, source hfile must be compressed.
- Table Split: In case, Source and Peer cluster have different split points for table regions, then hfile will be split before loading into peer cluster. It will slow down the replication process. To avoid this, Source and Peer cluster table should have same split points.
- Bandwidth control: The RPC call from hbase contains only hfile Paths. So there is no need to control the data limit in hbase. The main network data traffic is generated by copy of hfiles from one HDFS cluster to another.

# 5 Backward Compatibility

- No changes in existing public APIs.
- Replication queue structure for WAL files will be changed.
  Currently WALs are under the rs/peer node, now it will be moved to rs/peer/wals node. On startup, RS will process the old entries first.

# 6 Security

- Peer cluster will require Read permission for active HDFS cluster.
- Other security configuration remains same as existing.
- Audit Logs: Audit logs of Peer cluster RS will contain the replication audit info.

# 7 Metrics

All below metrics are at active RS level.

- Number of hfiles in queue
- Number of hfiles shipped

- Age of last hfiles shipped

All below metrics are at peer RS level.

- Number of hfiles applied
- Age of last hfiles applied

# 8 Configuration

All below configurations are at RS level.

- hbase.bulkload.replication.enabled=false/true [ Default : false]
- hbase.bulkload.replication.hfilerequest.count=<< Long >> [ Default: 100]

  Number of hfiles to grouped in single replication request.
- hbase.bulkload.replication.hfilerequest.waittime=<<Integer>> [Default: 30]

  Wait interval in seconds before sending replication request.
- hbase.bulkload.replication.hfile.queuesize=<< Long >> [Default: Max Long Value] Max number of hfiles in queue.

# 9 Interfaces (API)

No external interfaces