ArchitectureAntiEntropy

Anti-entropy Overview

AntiEntropyService generates MerkleTrees for column families during major compactions. These trees are then exchanged with remote nodes via a TreeRequest/TreeResponse conversation, and when ranges in the trees disagree, the 'org.apache.cassandra.streaming' package is used to repair those ranges.

Tree comparison and repair triggering occur in the single threaded AE_SERVICE_STAGE.

The steps taken to enact a repair are as follows:

1. A major compaction is triggered either via nodeprobe, or automatically:
    - Nodeprobe sends TreeRequest messages to all neighbors of the target node: when a node receives a TreeRequest, it will perform a readonly compaction to immediately validate the column family.
    - Automatic compactions will also validate a column family and broadcast TreeResponses, but since TreeRequest messages are not sent to neighboring nodes, repairs will only occur if two nodes happen to perform automatic compactions within TREE_STORE_TIMEOUT of one another.

2. The compaction process validates the column family by:

- Calling getValidator() (which can return a NoopValidator if validation should not be performed),
- Calling IValidator.prepare(), which samples the column family to determine key distribution,
- Calling IValidator.add() in order for every row in the column family,
- Calling IValidator.complete() to indicate that all rows have been added.
    - If getValidator decided that the column family should be validated, calling complete() indicates that a valid MerkleTree has been created for the column family.
    - The valid tree is broadcast to neighboring nodes via TreeResponse, and stored locally.

3. When a node receives a TreeResponse, it passes the tree to rendezvous(), which checks for trees to rendezvous with / compare to:

- If the tree is local, it is cached, and compared to any trees that were received from neighbors.
- If the tree is remote, it is immediately compared to a local tree if one is cached. Otherwise, the remote tree is cached in case a local tree is generated within TREE_STORE_TIMEOUT.
- A Differencer object is enqueued for each comparison.

4. Differencers are executed in AE_SERVICE_STAGE, to compare the two trees.

- If the trees disagree, the differencer will perform repair for the mismatched ranges via the io.Streaming api.

TODO

Repairs currently require 2 major compactions: one to validate a column family, and then another to send the disagreeing ranges.

One possible fix to this problem would be to use something like a Linear Bloom Filter to store a summary of every SSTable on disk, where each sub-bloom is partitioned using 'midpoint()' like the current MerkleTree. Then, to validate a column family, you could OR together the bloom filters for each SSTable, and send it to neighbors without performing a compaction.

`https://c.statcounter.com/9397521/0/fe557aad/1/|stats`