Tags in HBase - HBASE 8496

Ramkrishna S Vasudevan

Intel

1/8/2013

Tag methods in Cell Interface

Add the following methods to the Cell interface

```
/**
  * @return the first offset where the tags start in the Cell
  */
 int getTagsOffset();

 /**
  * @return the total length of the tags in the Cell.
  */
 short getTagsLength();

 /**
  * @return the entire tag byte array
  */
 byte[] getTagsArray();
```

Structure of tags

- **<2 byte tag length><1 byte type code><tag>**
- Every KV can have 1 or more tags.
- We need to provide some Tag Iterators to iterate the tags
- The Iterator must use the above tag structure to build this info.
- The reason for having the type is to support different type of tags like Visibility, ACL etc.
- CPs can be used to evaluate the tag based on the type and use new filters that work on Tags

How to specify tags

- Operation Attributes of Puts/Scans
- Use Put.add(KeyValue) where the KeyValue will be the one which understands tag.
- If we use Operation Attributes and make the user specify the tag type then we can use different CPs for different cases and handle the validation. Operation Attributes will not allow to specify the tags per KV and will allow to specify the tags for that set of Mutations.
- When we go with CP based approach client side validation of Tags is not possible.

HFileV3

- In order to support Tags and add the Tags in the KVs written to Hfile we would create a new version of Hfile which would be V3 and this just adds the taglength and tag after the keyvalue part.
- **<keylength><valuelength><keyarray><valuearray><taglength><tagarray>**

- The HFileWriterV3 extends HFileWriterV2 and will be able to persist the Tags in the Hfiles created with version V3.
- The HFileReaderV3 extends HFileReaderV2 and will be able to understand Tags in the read flow.
- The idea behind V3 is to have minimum impact to the existing usecases without Tags.
- The DataBlockEncoders uses the Encoding/Decoding context to track the memstoreTS and tags and the Encoding algos track memstoreTS and tags using these contexts.

HFileV3

- The HFileBlock does not go for any change except that like memstoreTS, tags would be tracked.
- Pls note that we would not be persisting any tag related information on the HFileBlock. So is the case with memstoreTS.

Encoder changes

- Based on the Encoding/Decoding context state the Encoder and decoding logic of the algos would handle tags.
- All encoding algos except PrefixTree Codec just persists the tags if found in the Byte buffer. Does not apply any special logic on the tags.
- PrefixTreeEncoders works like how qualifer's prefix tree works.
- Here again PrefixTreeEncoder will behave the older way for cases where there are no tags added.

Making Tags Optional

- In the current existing design memstoreTS works in an optional way, wrt how the writer writes it and how the reader decodes it.
- Similarly tags also can be made optional when using HFileVersion V3.
- When V3 is selected for Hfile, during flush we would always write the tagLength in the Hfile and would track the max tag length.
- The max tag length would be written to the FileInfo of the Hfile.
- When the reader is created for compacting the flushed Hfiles, the max tag length from the file info would determine we need to write the tags in the compacted file. If max tag length is 0 then tags need not be written.
- This would ensure that when there are no tags there is no performance impact.
- Currently note that the tags length is treated as **short.**

Client side Changes

- Support Tags in Put.
- Add additional options to Performance Evaluation tool and LoadTestTool to test with Tags.
- ImportTSV tool has to be changes to support tags for bulk loading.

TODOs

- Add the feature of Tag dictionary to the Encoding Algos except for Prefix Tree algo which already treats tags as qualifiers.
- WAL compression can also use tag dictionary.
- Find out an efficient and performance oriented way of stripping out Tags from KVs.
- Add a feature in such way that we have a provision to configure different RPC codecs on the client and Server side. So that for security features the RPC codec could stripe of the Tag section.
- Support Tags in Append and increment operation.