

Tag support in HBase

In order to support tags in HBase we have tried out two options

- ➔ Adding Tags in the same KV byte buffer
- ➔ Subclassing KV as TaggedKV and making Tags inmemory to the KV

Tag appended to KV byte buffer	Tagged KV approach (Tag as inmemory to KV)
Uses the same kv byte buffer. Added to the end of the current kv structure	TaggedKV subclass KV and has a Tag class that has a byte[], tagoffset and taglength representing the tags. Has a setTags() and remains in memory like memstoreTS
Tags uses HFileWriterV3/ReaderV3.	Tags uses HFileWriterV3/ReaderV3.
KV class gets changed to support passing Tags in constructor	No change to KV class.
KVs without tags are not affected including the byte[] remains same.	KVs without tags are not affected
No impact on the KV size in memstore both with Tag and without Tag except for the addition to the KV buffer due to tags	For cases without tags there are no impact but with tag the Tag class adds some additional overhead (32 byte) per KV + tagLength
WriterV3 writes Tag and taglength as integer	WriterV3 writes Tag and tagLength as vint
ReaderV3 understands Tag and creates KVs with Tags	ReaderV3 understands Tags and creates KV and sets Tags in memory to every KV. But every KV created is TaggedKV inside V3. No typecast required.
V3 can support KVs without tags also. Avoids reading/writing tags for HFiles without Tags during compaction	V3 can support KVs without tags also. Avoids reading/writing tags for HFiles without Tags during compaction
Putting tags into the kv byte[] makes it hard to filter them out for the security cases without copying	As it is inmemory this would be faster and cheaper, particularly if we have system tags later in the future.
No impact on RPC and WAL cell codec as the KV bytes are always serialized and deserialized fully.	RPC and WAL codec has to be changed so that the in memory tags are persisted.
No problem in compatibility	WAL and RPC side there may be compatibility issues if not committed to 0.96.0. So need to handle it separately to avoid this.
No impact on V2 formats	No impact on V2 formats

Note:

Client side changes needed irrespective of the above mentioned format.

The things indicated in red are the points that are to be looked into for the design decision