

Design Considerations

Kevin Liew

May 17, 2016

1 RPC

How are ResultSets transferred? Batches contained in Thrift objects? Contiguous batches of TRowSet.

ie.

```
ResultSet{
  TRowSet batch1{
    TColumn1{row1, row2},
    TColumn2{row1, row2},
    columnCount = 2
  }
  TRowSet batch2{
    TColumn1{row3, row4},
    TColumn2{row3, row4},
    columnCount = 2
  }
}
```

except that the columns are in 'binaryColumns' rather than a list of TColumn after HIVE-12049

2 Interface

We can provide a default compressor list for beeline. But for other apps using the JDBC driver, we shouldn't modify connection parameters. Should we make new versions of the JDBC driver always send all compressors in the classpath (as a handshake rather than a connection string parameter) so that performance gains are seen o-o-t-b without configuration from the user?

3 Compressor-Decompressor

How do we handle the case where the compressor throws an exception while handling a batch? Do we restart serialization and serialize the batch as an uncompressed column? What do we do with prior batches that were already serialized? We may have to restart serialization of the whole result set. Is there a better solution?

Where does decompression occur in the client? Will we operate on (Encoded)ColumnBasedSet or directly on TRowSet? Which files should I look at? We probably get a TRowSet and deserialize that to (Encoded)ColumnBasedSet from which we decompress batch(es).

hive.server2.thrift.resultset.max.fetch.size: "Max number of rows sent in one Fetch RPC call by the server to the client." If the client receives batch-by-batch, then we don't need to store the batch size in TEnColumn because the client will receive and decompress each batch separately instead of receiving one huge blob with all batches? So does the client receive the batches separately or in one blob?