ArchitectureSSTable

SSTable Overview

DRAFT. Notes on documenting how SSTables work in Cassandra (data format, indexing, serialization, searching)

SSTables have 3 separate files created, and are per column-family.

1. Bloom Filter
2. Index
3. Data

When adding a new key to an SSTable here are the steps it goes through. All keys are sorted before writing.

1. Serialize Index (ColumnIndexer.serialize(!IIterableColumns columns, DataOutput dos))
    a. Sort columns for key
    b. Serialize columns bloom filter
        i. Loop through columns and subcolumns that make up for column family
            1. Build sum for columnCount by column getObjectCount (includes getting subcolumn counts for super columns)
            2. Create bloom filter with column count
            3. Loop through columns (again) and add column name to bloom filter
                a. If super column detected, loop through subcolumns and add column name
        ii. Write bloom filter hash count (int)
        iii. Write serialized bloom filter length (int)
        iv. Write serialized bytes of bloom filter
    c. Start indexing based on column family comparator
        i. If columns empty write integer zero, return
        ii. Iterator over all columns creating a collection of IndexHelper.IndexInfo objects each IndexInfo representing at most getColumnIndexSize() worth of data (default is 64KB: Value from yaml's column_index_size_in_kb)
            1. Construct each new IndexInfo that consists of first and last columns visited that fit in the index size limit
        iii. Write size of indexSizeInBytes (int)
        iv. Serialize each IndexInfo object - (firstname is first column name visited in block, and lastname is the last column name visited)
            1. Write byte firstname - (length >> 8) & 0xFF
            2. Write byte firstname - (length & 0xFF)
            3. Write byte firstname
            4. Write byte lastname - (length >> 8) & 0xFF
            5. Write byte lastname - (length & 0xFF)
            6. Write byte lastname
            7. Write long startPosition
            8. Write long endPosition - startPosition
2. Serialize Data (ColumnFamilySerializer.serializeForSSTable(ColumnFamily columnFamily, DataOutput dos)
    a. Write columnFamily localDeletionTime (int)
    b. Write columnFamily markedForDeleteAt (long)
    c. Sort columns
    d. Write the number of columns (int)
    e. Determine Column Serializer and Serialize Column
        i. Determine length of column name as length
        ii. Write byte - (length >> 8) & 0xFF
        iii. Write byte - length & 0xFF
        iv. Write byte of column name
        v. Write boolean isMarkedForDelete
        vi. Write long timestamp
        vii. Write column value length (int)
        viii. Write column value as byte
3. Write to SSTable Data File
    a. Write out row key in UTF, this is based on partitioner
        i. Random Partitioner
            1. key token + DELIMITER + key name
            2. Delimiter is colon
    b. Write size of row value (int)
    c. Write byte of row value
4. Write SSTable Bloom Filter and SSTable Index
    a. Add to bloom filter disk key based on partitioner
        i. Random Partitioner
            1. key token + DELIMITER + key name
            2. Delimiter is colon
    b. Write disk key to SSTable Index file (UTF)
    c. Write file position before (Write to SSTable Data File) (int)