BigTable and Accumulo

CMSC 461

Michael Wilson

BigTable

- This was Google's original distributed data concept
- Key value store
- Meant to be scaled up into the petabyte range
- Used by many of Google's online applications
  - Google Maps, Youtube,

BigTable storage

- There are many “tables” in a single deployment of BigTable
- Each “cell” in a table consists of three values
  - Row key (string)
  - Column key (string)
  - Byte array
  - Timestamp
- It takes nothing more than this

Tablets

- Tables are split into tablets
  - Tablets are subdivisions of the parent table
  - Spread across the machines present in a BigTable cluster
- When tablets become too large ~200 MB, they “split”
  - The bigger the table becomes, the more tablets you have

Metadata tables

- Spread amongst the regular tablets are META1 tablets
  - META1 tablets have information on where specific data is located
    - Ranges of data (row keys, column keys)
- META0 tablet
  - Special tablet that maintains the locations of the various META1 tablets

Scanning for data

- Queries in BigTable are done by row key and column key
  - These are pure strings – really just linearly searching through each tablet in order to find the appropriate strings
  - Metadata tables help you figure out which tablets are worth searching
- To impose some sort of structure or order on the various cells, you need to cleverly construct your row/column keys

Pre-splitting a table

- Tables can be set up with existing "split points"
  - If a developer knows that the data being inserted into the table will cause (or likely cause) a number of splits, split in advance
  - Splits can be costly
- Tables not big enough to trigger splits
  - If you have a lot of small data, it can be useful to split it yourself

Split sizes

- You can also control the split sizes of a table
  - Data is not very large, but you want it to grow in an intelligent way
  - Set the split sizes to a more reasonable number
  - Typically determined in bytes

Accumulo

- Accumulo is based on the Google's BigTable paper
- Accumulo does things slightly differently

Server roles

- Master
  - This is the equivalent of META0 and META1 combined
- Tablet server
  - This houses various tablets spread across a table
  - A tablet server can host many tablets
- Logger
  - Maintains writeahead logs

Server roles

- gc
  - Garbage collects across the various tablets
- monitor
  - Web page one can access to monitor various data about Accumulo

Cell break down

- Cells in Accumulo are similar, but there are differences
  - Row ID
  - Column Family
  - Column Qualifier
  - Column Visibility
  - Timestamp
  - Value

Cell level security

- Accumulo adds something that BigTable doesn't have by default
- Column visibility field
  - Allows you to assign "authorizations" to various cells in a table
  - You can issue the same query with different authorizations and get different results
  - Allows you to compartmentalize sensitive data from users who may not have access
    - RDBMSes have label security for a similar purpose

Duplicate behavior

- Accumulo is capable of maintaining duplicates due to keeping track of the timestamp
  - It can be configured to behave in particular ways
  - By default, rows are overwritten with the most recent timestamp version

Iterators

- Accumulo also has “iterators” which can be applied to tables
  - Allow for arbitrary code to be executed on tables exhibiting certain properties
  - Example: SummingCombiner
    - Can establish that the values of all rows with the same row ID and column family should be summed together
- Can be applied to whole tables or can be issued as part of a standalone scan

MapReduce and Accumulo

- One can pass an Accumulo table as input to a MapReduce job, or use an Accumulo table as output from a MapReduce job
  - The number of tablets you have = the number of mappers you have
    - One tablet processed by one mapper