Subsection{apache that area no longer useful for accumulo

Business, Management

subsection{Apache Accumulo}Apache accumulo belongs to wide column store family and it is developed based on Java. It provides eventual consistency and availability. Data can be accessed through declarative query language, API based and REST/HTTP based queries. Language support is very minimal compared to other databases. It is schema less for database, but it has physical schema for key-value pair.

Accumulo stores data in memory and it also provides other reliable options like HDFS (Hadoop File System). subsubsection{Data model}Accumulo data model looks very similar to key value pairs, but key is divided in to three sub elements. RowID specifies the row number for the specific value, and it is not unique. Column specifies the column name of the value. Timestamps represent the time when the value is stored. Accumulo sorts keys by element and lexicographically in ascending order and timestamps are sorted in descending order cite{misc13}. Column (value) again divided into three sub elements, Family which is used to group similar column names, Qualifier represents the column name itself and Visibility defines the rules for who can access this value.

subsubsection{Components of accumulo}extbf{Tablet server}Tablet server take care of read and write requests from client and stores the data initially in write-ahead log. Once the memory reaches it threshold, all data will be moved to new files in HDFS. extbf{Garbage collector}Accumulo process stores temporary files and objects in HDFS and garbage collector periodically checks those garbage chunks that area no longer useful for accumulo process and delete them.

For robustness, multiple garbage collectors possibly run in standby mode. If current garbage collector fails, then based on election a new garbage collector will be assigned. extbf{Master}Master plays an important role in the accumulo architecture. It is responsible for detecting TabletServer failure and assign the tablets to different Tabletserver.

Master assigns the tablets to the Tabletserver and unload the tablets from Tabletserver when necessary. Master is also responsible for recovery management if a TabletServer fails. To ensure availability, multiple masters can be initiated and one master will be chosen based on election process. Other masters act as a backup servers. extbf{Monitor}Accumulo monitor is a web application which comes with accumulo package and it helps to constantly check the wealth information of an accumulo server like read/write rates, cache hit/miss rates, scan rate, active/queued compaction’s. For debugging accumulo database, monitor is the first entry point to find problems.

Like other components have backup option, we can have multiple monitor instance which can be used in case of monitor failure. extbf{Client}Accumulo includes a client library that is linked to every application and it contains logic for finding servers and communicating with them to write and read key-value pairs. cite{misc13}extbf{Fault tolerance}In case of TabletServer failures, master automatically moves the tablets to other TabletServer. If any data stored in to WAL during the failure, fault tolerance system also moves that data to newly created TabletServer, to ensure consistency.

subsubsection{Replication}Replication in accumulo copy data automatically to other systems because for disaster recovery, high availability or geographic locality. cite{misc14}The instance which is handling the current read/write request is called as primary/local instance and the replicated servers are called as peers. Accumulo is eventually consistent in cluster mode, but it is strongly consistent in single instance mode. Using ZooKeeper, we can lock the TabletServer during replication of files to peer and later Master and Garbage Collector remove records from meta data and replication tables and files from HDFS respectively.