Summary

This review covers systems that provide relational access to data hosted in the cloud. Three solutions are looked at: Microsoft's Cloud SQL Server, Pig, and H-Store.

Microsoft Cloud SQL Server is a distributed version of SQL Server. In fact, it uses SQL Server as the underlying query processor. The system limits transactions to a single table group (if keyless) or row group (if keyed), which cannot exceed the size of a node. The reason for limiting transactions is to eliminate the need for two phase commits and distributed locking. Partitioning can be controlled with the use of a partitioning key, allowing data accessed at the same time from multiple tables to be included in the same partition. The system architecture consists of 5 main parts. There is a protocol gateway that accepts incoming client connections and forwards queries to the necessary node. The distributed fabric monitors nodes for failures and joins as well as performs leadership elections. A global partition manager keeps track of the location, state, and history of each partition replica. Nodes communicate with it when migrating, splitting, and changing the primary status of a replica. A deployment manager allows the system to be upgraded without needing to shut down the cluster.

Pig allows relational access to data when updates are not required. This is particularly useful when ad-hoc analysis is desired. One of the benefits of Pig is that it can be used directly on raw data files without needing to rearrange the data into proper tables and import the data into a database. Pig converts scripts written in Pig Latin to MapReduce jobs, enabling straightforward relational access. Pig Latin contains operations commonly available in single machine RDBMSs, but which are typically left out of distributed solutions due to high message overhead and complicated locking schemes. This includes operations like JOIN.

H-Store takes a different approach to transaction management. The expected operations are submitted when the cluster launched. These pre-defined stored procedures contain control code and parameterized SQL statements. The stored procedures are used to determine the physical partitioning and replication scheme. For example, if certain read-only data is accessed often, the data will have increased replication to eliminate network traffic. H-Store expects that data partitions will fit into available main memory. It uses this expectation to its advantage by assigning a single thread to each partition, eliminating some locking requirements in the process. Instead, transactions accessing the partition are queued and are executed by the thread in order, one after the other. There are two types of transactions in H-Store. Single-partition transactions do not require locks and can be submitted directly to the desired partition. Multi-partition transactions, on the other hand, require a global ordering. This can be achieved either with the use of a global coordinator or two-phase locking. The scheme that provides higher throughput depends on the percentage of multi-partition transactions.

Review

Cloud SQL Server has many similarities to BigTable. In the case of a keyed table group, Cloud SQL server limits transactions to one row group. A single row group includes all of the rows with the same partition key, which may be from different tables. In BigTable, a read or write to a row is atomic, no matter the number of columns that are read or written. With the use of column families, it is possible to store relational data under a single row in BigTable, as is done with anchors. As far as transactions are concerned, the row key and the partitioning key are effectively the same. The serving scheme has similarities as well. In Cloud SQL server, only the primary replica of a partition is allowed to process queries. This eliminated to risk of reading stale data or losing isolation. The primary then sends after-images of updated data to the secondary replicas, requiring a quorum of them to acknowledge the updates. In BigTable, each tablet is assigned to one tablet server at a time. Only this tablet server is allowed to read and write to the tablet. BigTable does not need to replicate tablet servers, however, since committed updates are automatically replicated by GFS.