BigTable Characteristics

This lesson will explore some miscellaneous characteristics of BigTable.

We'll cover the following

BigTable performance#

Here are a few reasons behind BigTable's performance and popularity:

- **Distributed multi-level map**: BigTable can run on a large number of machines.
- **Scalable** means that BigTable can be easily scaled horizontally by adding more nodes to the cluster without any performance impact. No manual intervention or rebalancing is required. BigTable achieves linear scalability and proven fault tolerance on commodity hardware.
- **Fault-tolerant and reliable**: Since data is replicated to multiple nodes, fault tolerance is pretty high.
- **Durable**: BigTable stores data permanently.
- **Centralized**: BigTable adopts a single-master approach to maintain data consistency and a centralized view of the state of the system.
- **Separation between control and data**: BigTable maintains a strict separation between control and data flow. Clients talk to the Master

for all metadata operations, whereas all data access happens directly between the Clients and the Tablet servers.

Dynamo vs. BigTable#

Here is the comparison between Dynamo and BigTable:

	Dynamo	BigT
Architecture	**Decentralized** Every node has same set of responsibilities	**Centa** Master handles serv handle re
Data Model	**Key-value**	**Multidimensio**
Security	X	Access rights at c
Partitioning	**Consistent Hashing** Each node is assigned to a random position on the ring	**Tab** Each table is broke range of rows
Replication	**Sloppy Quorum** Each data item is replicated to 'N' number of nodes	**GFS Chunk** Data is stored in GF broken into chunks are replicated to
CAP	**AP**	**C**
Operations	**By key**	**By key**
Storage	**Plug-in**	**SSTable**
Memberships and failure detection	**Gossip based protocol**	**Handshakes initia**

Datastores developed on the principles of BigTable#

Google's BigTable has inspired many NoSQL systems. Here is a list of a few famous ones:

HBase: HBase is an open-source, distributed non-relational database modeled after BigTable. It is built on top of the Hadoop Distributed File System (HDFS).

Hypertable: Similar to HBase, Hypertable is an open-source implementation of BigTable and is written in C++. Unlike BigTable, which uses only one storage layer (i.e., GFS), Hypertable is capable of running on top of any file system (e.g., HDFS (https://en.wikipedia.org/wiki/Apache_Hadoop#HDFS), GlusterFS (https://en.wikipedia.org/wiki/Gluster#GlusterFS), or the CloudStore (https://en.wikipedia.org/wiki/CloudStore)). To achieve this, the system has abstracted the interface to the file system by sending all data requests through a Distributed File System broker process.

Cassandra: Cassandra is a distributed, decentralized, and highly available NoSQL database. Its architecture is based on Dynamo and BigTable. Cassandra can be described as a BigTable-like datastore running on a Dynamo-like infrastructure. Cassandra is also a wide-column store and utilizes the storage model of BigTable, i.e., SSTables and MemTables.

← Back

BigTable Refinements

Next →

Summary: BigTable

Mark as Completed

Report an Issue