HBase vs Cassandra

(What to choose?)

Most of the current generation software application still depend on relational databases such as Oracle, PostgresSql for their online storage needs. There is equal number of applications where developer community lean towards column-oriented databases. Especially, when they need high distribution of data. In other words, scaling horizontally is more important for their organization.

The most famous options for column-based databases are HBase and Cassandra. Data Engineers & Software Developers often get confused when its time choose one against another. HBase and Cassandra both are well known column-based databases. Both work on distributed systems architecture. Both designs support highly available. With right configurations and cluster sized both can reach 100k records/sec easily. Then how do we choose from these options? Let us discuss the differences and choosing strategy in follow sections.

Note: This document does not talk about optimizing the HBase or Cassandra. Rather helps in an architectural decision and understand which would work better for your use case.

Write Strategy:

HBase & Cassandra considers a record persisted as soon as the record is written to Write Ahead Logs (WAL) and persisted to mem-tables.

Write Ahead Logs concept allow the column-based databases write the data to disk sequentially without moving disk head. This concept ensure data that reached server never lost. In case of system failure, the WALs are replayed to persist.

Memtables as name suggests, data is store in memory as soon as it arrives a server. The data write then immediately acknowledged. The tools achieve high write speeds with the help of memtables.

WALs helps with write recovery and memtables helps with write velocity.

Speed:

HBase: As soon as a write request is received, HBase writes the data directly to WAL. Then this data is also persisted in memory starting fill the in-memory buffer. This buffer is then spilled to disk. Both WALs and memory table spills are written to disk at high speeds. These speeds are achieved by writing a block of info at any time. Disk seeks are optimized by writing the data sequentially and as blocks.

Cassandra: Cassandra on the other hand follows similar strategy. The recent improvements in Cassandra writes both WAL and mem-tables simultaneously. This reduces the write latency giving the Cassandra better score in write strategy.

If write speed/write performance is your priority, Cassandra could make a good choice.

Read Strategy:

Column oriented databases are faster when data is accessed using key. Reads other than key based could potentially result in full table scan. Both tools have secondary strategies to avoid full tables scans. These topics are out of scope for this document.

Consistency:

HBase: Since HBase has master based architecture, the data when requested always read from the corresponding partition's master node. The most recent write for this record would have happened 100% percent to the same node. This ensure that the reads in HBase are always most recent and consistent.

Cassandra: Cassandra's master/server less architecture, could redirect write and read of the same record to different nodes. While there are other advantages of this architecture, consistency is not one of them. Unless otherwise very well-tuned, achieving read consistency could be challenge with-out compromising on performance. Cassandra has read settings to achieve consistency to a level but, full consistency is not promised.

If your application depends on high read consistency, HBase would be right choice.

Disaster Recovery

In case of disaster occurs at your database location, you would need a secondary database location that syncs your data constantly. Disaster recovery is part of every company's strategy these days.

Cassandra is capable of replicating data in different data centers almost real time. HBase lacks some of these capabilities.

Disaster recovery and cross datacenter replication makes Cassandra a great tool with of operational excellence.

Support & Community

Developer Community
Some of us agree that Stackoverflow could be a good place to check active dev community. Stackoverflow has 6500+ questions for HBase and 17500+ questions for Cassandra. This is indication that both tools have very good developer community to help out with day-to-day development issues.

Enterprise Support
Sometime the developer community will not be enough, especially when you are working with high velocity applications thing could go out of control often. You would need expert help 24/7. Enterprise level support from third party community could come in handy in such situations.

- HBase has enterprise level support from Cloudera.
- Cassandra has support from Datastax

If you are looking for a good developer community and cannot afford enterprise level support, then Cassandra could make good choice.

Conclusion

I would like to conclude my technology review for two different kinds of audience. Beginner, Intermediate level experience in using column-oriented databases. I'm not giving any conclusion for experts as they are probably not reading this document but rather, looking for documentation on fine tuning their applications ☺

Beginner
Cassandra solves most of the day-to-day problems you would face during the development. It also supports wide variety of query languages SQL, CQL etc. Cassandra also has very good connectors supported by community to transfer the data from one type of database to another. For beginners, Cassandra would make better choice simply due to the ease of use.

Intermediate
As discussed in previous sections Cassandra is a clear winner several categories. If high consistency is not of your concern, Cassandra comes with bundle of features and optimizations. Cassandra is could help you with most of your use cases such as better write speeds, better read speeds, cross datacenter replication, better support etc. It is suggested to choose HBase if consistency is your highest priority.