

	Blogs	Videos
--	-------	--------

Interview Questions
[\(https://www.edureka.co/blog/interview-questions/\)](https://www.edureka.co/blog/interview-questions/)





Insights on HBase Architecture

Priyanka (<https://www.edureka.co/blog/author/priyanka/>)
 Dec 19, 2016

Recommended by 12 users

(<https://www.facebook.com/sharer/sharer.php?u=https://www.edureka.co/blog/insights-on-hbase-architecture>)
[Twitter](#) [LinkedIn](#) [Google+](#) [Print](#) [Email](#)



 Add to Bookmark (<https://www.edureka.co/blog/insights-on-hbase-architecture>)
 Email this Post  13.1K  (<https://www.edureka.co/blog/insights-on-hbase-architecture#comments-wrapper>) 1 (https://www.edureka.co/blog/insights-on-hbase-architecture#disqus_thread)

(<https://www.edureka.co/blog/insights-on-hbase-architecture>)

HBase – The Basics:

HBase is an open-source, NoSQL, distributed, non-relational, versioned, multi-dimensional, column-oriented store which has been modeled after Google BigTable that runs on top of HDFS. "NoSQL" is a broad term meaning that the database isn't an RDBMS which supports SQL as its primary access language. But there are many types of NoSQL databases and Berkeley DB is a good example of a local NoSQL database, whereas HBase is very much a distributed database.

HBase provides all the features of Google BigTable. It began as project by Powerset to process massive amounts of data for natural language search. It was developed as part of Apache's Hadoop project and runs on top of HDFS (Hadoop Distributed File System). It provides fault-tolerant ways of storing large quantities of sparse data. HBase is really more a "Data Store" than "Data Base" because it lacks many of the features available in RDBMS, such as typed columns, secondary indexes, triggers, and advanced query languages, etc.

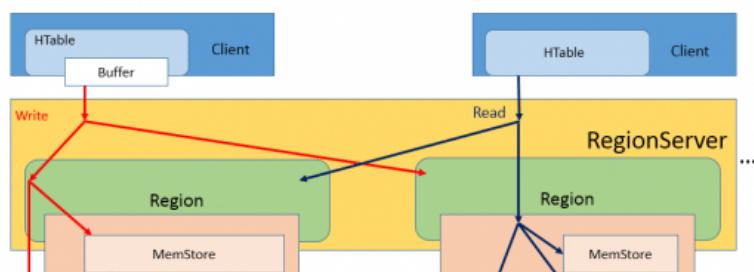
In the Column-Oriented databases, data table is stored as sections of columns of data rather than as rows of data. The Data model of column oriented database consists of Table name, row key, column family, columns, time stamp. While creating tables in HBase, the rows will be uniquely identified with the help of row keys and time stamp. In this data model the column family are static whereas columns are dynamic. Now let us look into the HBase Architecture.

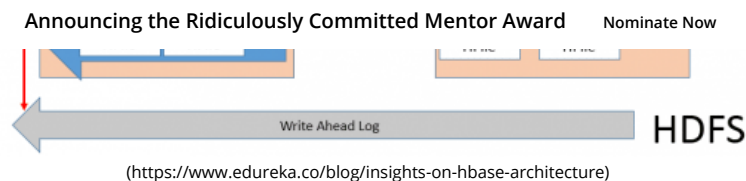
When to go for HBase?

HBase is a good option only when there are hundreds of millions or billions of rows. HBase can also be used in places when considering to move from an RDBMS to HBase as a complete redesign as opposed to a port. In other words, HBase is not optimized for classic transactional applications or even relational analytics. It is also not a complete substitute for HDFS when doing large batch MapReduce. *Then why should you go for HBase?? If your application has a variable schema where each row is slightly different, then you should look at HBase.*

HBase Architecture:

The following figure clearly explains the HBase Architecture.





In HBase, there are three main components: **Master, Region server and Zoo keeper**. The other components are **Memstore, HFile and WAL**.

As HBase runs on top of HDFS, it utilizes the Master-Slave architecture in which the HMaster will be the master node and the Region Servers are the slave nodes. When the client sends a write request, HMaster gets that request and forward it to the respective Region Server.

Region Server:

It is a system which acts similar to a data node. When Region Server (RS) receives write request, it directs the request to specific Region. Each Region stores set of rows. Rows data can be separated in multiple column families (CFs). Data of particular CF is stored in HStore which consists of Memstore and a set of HFiles.

What does Memstore do?

Memstore keeps track of all the logs for the read and write operations that has been performed within that particular region server. From this we can say that is acting similar to a name node in Hadoop. Memstore is an in-memory storage, hence the Memstore utilizes the in-memory storage of each data node to store the logs. When certain thresholds are met, Memstore data gets flushed into HFile.

The key purpose for using Memstore is the need to store data on DFS ordered by row key. As HDFS is designed for sequential reads/writes, with no file modifications allowed, HBase cannot efficiently write data to disk as it is being received: the written data will not be sorted (when the input is not sorted) which means not optimized for future retrieval. To solve this problem, HBase buffers last received data in memory (in Memstore), "sorts" it before flushing, and then writes to HDFS using fast sequential writes. Hence, HFile contains a list of sorted rows.

Every time Memstore flush happens one HFile created for each CF and frequent flushes may create tons of HFiles. Since during reading HBase will have to look at many HFiles, the read speed can suffer. To prevent opening too many HFiles and avoid read performance deterioration, HFiles compaction process is used. HBase will periodically (when certain configurable thresholds are met) compact multiple smaller HFiles into a big one. Obviously, the more files created by Memstore flushes, the more work (extra load) for the system. Added to that, while compaction process is usually performed in parallel with serving other requests and when HBase cannot keep up with compacting HFiles (yes, there are configured thresholds for that too), it will block writes on RS again. Like we discussed above, this is highly undesirable.

We cannot be sure that the data will be persistent throughout in Memstore. Assume that a particular datanode is down. Then the data that resides on that data node's memory will be lost.

To overcome this problem, when the request comes from the master it written to WAL as well. WAL is nothing but *Write Ahead Logs* which resides on the HDFS, a permanent storage. Now we can make sure that even when if the data node is down the data will not be lost i.e. we have the copy of all the actions that you are supposed to do in the WAL. When the data node is up it will perform all the activities again. Once the operation is completed, everything is flushed out from Memstore and WAL and is written in HFile in order to make sure that we are not running out of memory.

Let us take a simple example that I want to add row 10 then that write request comes in, it says it gives all the meta data to the Memstore and WAL. Once that particular row is written into HFile everything in Memstore and WAL is flushed out.

Zoo Keeper:

HBase comes integrated with Zoo keeper. When I start HBase, Zoo keeper instance is also started. The reason is that the Zoo keeper helps us in keeping a track of all region servers that are there for HBase. Zoo keeper keeps track of how many region servers are there, which region servers are holding from which data node to which data node. It keeps track of smaller data sets where Hadoop is missing out. It decreases the overhead on top of Hadoop which keeps track of most of your Meta data. Hence HMaster gets the details of region servers by actually contacting Zoo keeper.

Got a question for us? Mention them in the comments section and we will get back to you.

Related Posts:

Get started with Big Data and Hadoop (<https://www.edureka.co/big-data-and-hadoop>)

Helpful Hive Commands (<https://www.edureka.co/blog/hive-commands-with-examples>)

All you need to know about Hadoop (<https://www.edureka.co/blog/hadoop-tutorial/>)

Announcing the Ridiculously Committed Mentor Award

Nominate Now



(<https://plus.google.com/>)

f

Twitter

in

G+

(<https://www.facebook.com/share?u=https://www.edureka.co/blog/insights-on-hbase-architecture>)

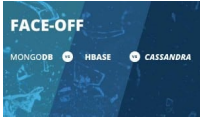
Share on

Got your brain cells running?
Stay tuned to latest technology updates


Enter your Email Address

SUBSCRIBE

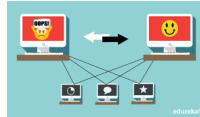
Related Posts




Face Off: MongoDB Vs HBase Vs Cassandra
4.8K



Top Hive Commands with Examples in HQL
174.1K



How to Set Up Hadoop Cluster with HDFS High Availability
46.9K



Sample HBase POC
13.2K

(<https://www.edureka.co/blog/mongodb-vs-hbase-vs-cassandra/>)

(<https://www.edureka.co/blog/top-hive-commands-with-examples/>)

(<https://www.edureka.co/blog/how-to-set-up-hadoop-cluster-with-hdfs-high-availability/>)

(<https://www.edureka.co/blog/sample-hbase-poc/>)

Browse Categories

- Big Data NoSQL (<https://www.edureka.co/blog/category/big-data-nosql/>)
- Blockchain (<https://www.edureka.co/blog/category/blockchain/>)
- Business Intelligence (<https://www.edureka.co/blog/category/business-intelligence/>)
- Cloud Computing (<https://www.edureka.co/blog/category/cloud-computing/>)
- Cyber Security (<https://www.edureka.co/blog/category/cyber-security/>)
- Deep Learning (<https://www.edureka.co/blog/category/deep-learning/>)
- Finance (<https://www.edureka.co/blog/category/finance/>)
- Frameworks (<https://www.edureka.co/blog/category/frameworks/>)
- Marketing (<https://www.edureka.co/blog/category/marketing/>)
- Mobile Development (<https://www.edureka.co/blog/category/mobile-development/>)
- Operations (<https://www.edureka.co/blog/category/operations/>)
- Programming (<https://www.edureka.co/blog/category/programming/>)
- Project Management (<https://www.edureka.co/blog/category/project-management/>)
- Robotic Process Automation (<https://www.edureka.co/blog/category/robotic-process-automation/>)
- Success Story (<https://www.edureka.co/blog/category/success-story/>)
- Systems & Architecture (<https://www.edureka.co/blog/category/systems-architecture/>)
- Systems Engineering (<https://www.edureka.co/blog/category/systems-engineering/>)

Announcing the Ridiculously Committed Mentor Award

[Nominate Now](#)[Loading... \(https://www.edureka.co/blog/insights-on-hbase-architecture/\)](#)

Comments

1 Comment

1 Comment <https://www.edureka.co/blog/> Rajiv Chaudhuri ▾ Recommend  Share

Sort by Best ▾



Join the discussion...



Samarjit Mahapatra • 5 years ago

good one

 ▾ • Reply • Share ▾ALSO ON [HTTPS://WWW.EDUREKA.CO/BLOG/](https://www.edureka.co/blog/)**Top 75 Talend Interview Questions and Answers for 2018**

2 comments • 6 months ago



Balázs Gunics — I found a few points that could be enhanced. OnComponent vs OnSubjobWhen using OnSubjob the previous function call is

What Is Microservices – Introduction To Microservice Architecture

1 comment • 7 months ago



Nadeem Inamdar — Good blog! Hate to be a grammar nazi .. but What ARE Microservices!

7 Reasons to Choose Edureka Online Courses

4 comments • 6 months ago






Gokhan Dedeoglu — My registered e-mail address Avatais gdedeoglu@gmail.com

Spring Boot Microservices: Building Microservices Application Using Spring Boot

97 comments • 4 months ago



Hardik — Hi, please share the source code at Avataidevk9383@gmail.com

 [Subscribe](#)  [Add Disqus to your site](#) [Add Disqus](#)  [Disqus' Privacy Policy](#) [Privacy Policy](#) [Privacy Policy](#) [Privacy Policy](#)Subscribe
to our newsletter

Enter your Email Address

SUBSCRIBE

Related Blogs

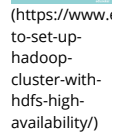
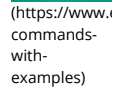
**Top 50 Hadoop Interview Questions You Must Prepare In 2018**
(<https://www.edureka.co/blog/interview-questions/top-50-hadoop-interview-questions-2016/>)

(https://www.edureka.co/blog/interview-questions/top-50-hadoop-interview-questions-2016/)

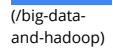
**Face Off: MongoDB Vs HBase Vs Cassandra**
(<https://www.edureka.co/blog/mongodb-vs-hbase-vs-cassandra/>)

(https://www.edureka.co/blog/mongodb-vs-hbase-vs-cassandra/)

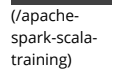
Nominate Now



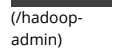
How to Set Up Hadoop Cluster with HDFS High Availability
(<https://www.edureka.co/blog/how-to-set-up-hadoop-cluster-with-hdfs-high-availability/>)



Big Data Hadoop Certification Training (/big-data-and-hadoop)



Apache Spark and Scala Certification Training (/apache-spark-scala-training)



Hadoop Administration Certification Training (/hadoop-admin)



Splunk Power User & Admin Certification Training (/splunk)

Follow us on

f  in 

(https://papers.ssrn.com/sol3/documents.cfm?abstract_id=3411111)

ms-


Learn on

the GO!

privacy-

(<https://itunes.apple.com/in/app/edureka/id1033145415?mt=8>)

emap)

 (https://play.google.com/store/apps/details?id=co.edureka.app)

(<https://play.google.com/store/apps/details?id=co.edureka.app>)

id=co.edureka.app)

"PMP®", "PMI®", "PMI-ACP®" and "PMBOK®" are registered marks of the Project Management Institute, Inc.
MongoDB®, Mongo and the leaf logo are the registered trademarks of MongoDB, Inc.