Introduction to Apache MapReduce and HDFS

Priyanka (https://www.edureka.co/blog/author/priyanka/) Dec 09,2015

(https://www.facebook.com/sharer.php?u=https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/) (https://twitter.com/share?url=https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/) (https://www.linkedin.com/shareArticle?mini=true&url=https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/) (https://plus.google.com/share?url=https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/)

Recommended by 46 users

Add to Bookmark (https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/) Email this Post 28.7K (https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/#comments-wrapper) 17 (https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/#disqus_thread)

(https://www.edureka.co/blog/introduction-to-apache-hadoop-hdfs/)

Introduction to Apache MapReduce and HDFS

Apache Hadoop has been originated from Google's Whitepapers:

1. Apache HDFS is derived from GFS (Google File System).
2. Apache MapReduce is derived from Google MapReduce
3. Apache HBase is derived from Google BigTable.

Though Google has only provided the Whitepapers, without any implementation, around 90-95% of the architecture presented in these Whitepapers is applied in these three Java-based Apache projects.

HDFS and MapReduce are the two major components of Hadoop, where HDFS is from the 'Infrastructural' point of view and MapReduce is from the 'Programming' aspect. Though HDFS is at present a subproject of Apache Hadoop, it was formally developed as an infrastructure for the Apache Nutch web search engine project.

To understand the magic behind the scalability of Hadoop from one-node cluster to a thousand-nodes cluster (Yahoo! has 4,500-node cluster managing 40 petabytes of enterprise data), we need to first understand Hadoop's file system, that is, HDFS (Hadoop Distributed File System).

What is HDFS (Hadoop Distributed File System)?

HDFS is a distributed and scalable file system designed for storing very large files with streaming data access patterns, running clusters on commodity hardware.

Though it has many similarities with existing traditional distributed file systems, there are noticeable differences between these. Let's look into some of the assumptions and goals/objectives behind HDFS, which also form some striking features of this incredible file system!

Assumptions and Goals/Objectives behind HDFS:

1. Large Data Sets:

It is assumed that HDFS always needs to work with large data sets. It will be an underplay if HDFS is deployed to process several small data sets ranging in some megabytes or even a few gigabytes. The architecture of HDFS is designed in such a way that it is best fit to store and retrieve huge amount of data. What is required is high cumulative data bandwidth and the scalability feature to spread out from a single node cluster to a hundred or a thousand-node cluster. The acid test is that HDFS should be able to manage tens of millions of files in a single occurrence.

2. Write Once, Read Many Model:

HDFS follows the write-once, read-many approach for its files and applications. It assumes that a file in HDFS once written will not be modified, though it can be access 'n' number of times (though future versions of Hadoop may support this feature too)! At present, in HDFS strictly has one writer at any time. This assumption enables high throughput data access and also simplifies data coherency issues. A web crawler or a MapReduce application is best suited for HDFS.

3. Streaming Data Access:

As HDFS works on the principle of 'Write Once, Read Many', the feature of streaming data access is extremely important in HDFS. As HDFS is designed more for batch processing rather than interactive use by users. The emphasis is on high throughput of data access rather than low latency of data access. HDFS focuses not so much on storing the data but how to retrieve it at the fastest possible speed, especially while analyzing logs. In HDFS, reading the complete data is more important than the time taken to fetch a single record from the data. HDFS overlooks a few POSIX requirements in order to implement streaming data access.

4. Commodity Hardware:

HDFS (Hadoop Distributed File System) assumes that the cluster(s) will run on common hardware, that is, non-expensive, ordinary machines rather than high-availability systems. A great feature of Hadoop is that it can be installed in any average commodity hardware. We don't need super computers or high-end hardware to work on Hadoop. This leads to an overall cost reduction to a great extent.

5. Data Replication and Fault Tolerance:

HDFS works on the assumption that hardware is bound to fail at some point of time or the other. This disrupts the smooth and quick processing of large volumes of data. To overcome this obstacle, in HDFS, the files are divided into large blocks of data and each block is stored on three nodes: two on the same rack and one on a different rack for fault tolerance. A block is the amount of data stored on every data node. Though the default block size is 64MB and the replication factor is three, these are configurable per file. This redundancy enables robustness, fault detection, quick recovery, scalability, eliminating the need of RAID storage on hosts and merits of data locality.

6. High Throughput:

Throughput is the amount of work done in a unit time. It describes how fast the data is getting accessed from the system and it is usually used to measure performance of the system. In Hadoop HDFS, when we want to perform a task or an action, then the work is divided and shared among different systems. So, all the systems will be executing the tasks assigned to them independently and in parallel. So the work will be completed in a very short period of time. In this way, the Apache HDFS gives good throughput. By reading data in parallel, we decrease the actual time to read data tremendously.

7. Moving Computation is better than Moving Data:

Hadoop HDFS works on the principle that if a computation is done by an application near the data it operates on, it is much more efficient than done far of, particularly when there are large data sets. The major advantage is reduction in the network congestion and increased overall throughput of the system. The assumption is that it is often better to locate the computation closer to where the data is located rather than moving the data to the application space. To facilitate this, Apache HDFS provides interfaces for applications to relocate themselves nearer to where the data is located.

8. File System Namespace:

A traditional hierarchical file organization is followed by HDFS, where any user or an application can create directories and store files inside these directories. Thus, HDFS's file system namespace hierarchy is similar to most of the other existing file systems, where one can create and delete files or relocate a file from one directory to another, or even rename a file. In general, HDFS does not support hard links or soft links, though these can be implemented if need arise.

Thus, HDFS works on these assumptions and goals in order to help the user access or process large data sets within incredibly short period of time!

After learning 'What is HDFS' in this write-up, further we will discuss the components of HDFS that form a significant part of the Hadoop cluster!

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)

Get Started with Comprehensive MapReduce (https://www.edureka.co/comprehensive-mapreduce-self-paced)

Get Started with MapReduce Design Patterns (https://www.edureka.co/mapreduce-design-patterns-sp)

Big Data Challenges (https://www.edureka.co/blog/big-data-challenges/)

Why BI professionals need to be skilled in Hadoop (https://www.edureka.co/blog/why-bi-professionals-need-to-be-skilled-in-hadoop/)

About Priyanka (8 Posts (https://www.edureka.co/blog/author/priyanka/))

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

f 🐦 in g+

(https://[illegible]
u=https:[illegible]
to- to- to- to-

Share on apache-hadoop-hdfs/) apache-hadoop-hdfs/) apache-hadoop-hdfs/) apache-hadoop-hdfs/)

Related Posts

Apache Hadoop HDFS Architecture
👁 61.6K

(https://www.edureka.co/blog/apache-hadoop-hdfs-architecture/)

10 Reasons Why Big Data Analytics is the Best Career Move
👁 251.4K

(https://www.edureka.co/blog/10-reasons-why-big-data-analytics-is-the-best-career-move)

5 Reasons to Learn Hadoop
👁 140.4K

(https://www.edureka.co/blog/5-reasons-to-learn-hadoop)

Comments

19 Comments

17 Comments https://www.edureka.co/blog/ Rajiv Chaudhuri

♡ Recommend 2 Share Sort by Best

Join the discussion…

Karthik Mannepalli • 2 years ago

I am sorry, typo in my previous question :
If the assumption is Write Once only and Read many times, does it mean, we cannot use HDFS for transactional data?

Reply • Share ›

Karthik Mannepalli • 2 years ago

If the assumption is Write Once only and Read many times, does it mean, we can use HDFS for transactional data?

Reply • Share ›

Khalid • 2 years ago

I expected to see here concise discussions on HDFS components: Namenode, Datanode and Secondary Namenode, but there isn't.

Reply • Share ›

Khalid • 2 years ago

Under 5. Data Replication and Fault Tolerance, it is pointed out the default HDFS block size being 64 MB. This is in fact true with Hadoop 1.x, but since Hadoop 2.0 it's been 128 MB. This blog was posted in May 2013 and apparently have not been updated since. So, I guess it'd be good if it was updated.

Reply • Share ›

Kumar • 4 years ago

Hi edureka, I want some resume formats for hadoop developer. Please forward if ur having that. Im new to this technology.
this is my mail id: akumarhadoop@gmail.com

Thanks in advance.
Kumar

Reply • Share ›

EdurekaSupport Mod → Kumar • 4 years ago

Hi Kumar, the sample resumes will be shared with you, by our support team only after you enroll for our 'Big Data & Hadoop' course.

• Reply • Share ›

Kushal ➦ EdurekaSupport • 2 years ago

Hi edureka team, Please share some resume formats for hadoop developer that relate to Big Data course. You can send me at kbvprasad@gmail.com ; Kushal.alester@gmail.com
@EdurekaSupport

• Reply • Share ›

Abhishek • 4 years ago

Can you please elaborate point #3?

"As HDFS is designed more for batch processing rather than interactive
use by users. The emphasis is on high throughput of data access rather
than low latency of data access. HDFS focuses not so much on storing the
data but how to retrieve it at the fastest possible speed, especially
while analyzing logs. In HDFS, reading the complete data is more
important than the time taken to fetch a single record from the data."

• Reply • Share ›

EdurekaSupport Mod ➦ Abhishek • 3 years ago

Hi Abhishek, batch processing is a technique which helps us to process the jobs without any manual information after submitting the job with required information (input, program name) . It keeps a track of jobs submitted and executes them in first come first serve fashion.
In Interactivity mode, User uses an interface to interact with system. It take the inputs from the user and output the result to the user using an
interface.
In Hadoop, once the job is submitted it takes the inputs and stores the results from/to the location we have given in the command. Hence
we call it as batch processing.
Throughput is nothing but the number of processed completed in a unit amount of time whereas Latency is the delay from the time we submit the job and get the desired outcome.
In Hadoop, we concentrate on increasing the throughput than decreasing the latency while processing a job as we need to retrieve the output at fast possible speed irrespective of size of data.
Hope this helps!

• Reply • Share ›

Karthik Mannepalli ➦ EdurekaSupport • 2 years ago

@EdurekaSupport - Doesn't increasing throughput reduce the latency? Both will go hand in hand right? Please correct me if I am wrong

• Reply • Share ›

Dr M. NAGABHUSHANA RAO • 4 years ago

Nice to see edureka blog, edureak is trying to spread knowledge on big data more. thank's to it's team for hardworking.

• Reply • Share ›

EdurekaSupport Mod ➦ Dr M. NAGABHUSHANA RAO • 4 years ago

Thanks a lot, Dr. Rao. Please feel free to go through our other blog posts as well.

• Reply • Share ›

Deepak Sharma • 4 years ago

Could you please elaborate on point #7 a bit more?

and also the line "Apache HDFS provides interfaces for applications to relocate themselves nearer to where the data is located"

• Reply • Share ›

EdurekaSupport Mod ➦ Deepak Sharma • 4 years ago

Hi Deepak,

Let us assume that we have a submitted a job and now jobtracker need to choose to which tasktracker node the job need to be allocated.

While assigning this job to the tasktracker, the jobtracker first finds out on which nodes the data resides and checks whether if that nodes are available to run the job/task. If yes, then it will assign the task to that
tasktracker nodes and then transfer the computed results to the other
nodes whichever are required. If not, it will assign that task to the
tasktracker nodes which are nearest to the nodes where the data resides. The reason why jobtracker tries to assign to the nodes
where the data resides because as the data in HDFS will be huge, it
may consume more amount of time due to network congestion/any other issues just to transfer the data instead of actual computation (the
actual thing which is important/required). Hence it is better to move the computed results (less data) instead of the actual data (huge data).

Hope this help!!!

^ | ⌄ • Reply • Share ›

Sushobhit Rajan • 4 years ago

Nicely Explained

^ | ⌄ • Reply • Share ›

EdurekaSupport Mod ➦ Sushobhit Rajan • 4 years ago

Thanks Sushobhit!!! Feel free to go through our other blog posts as well.

^ | ⌄ • Reply • Share ›

Gaurav Dighe • 5 years ago

Very nice information information about Hadoop. Keep up the good work.

Hope to see some more topics on DataFlow, Map Reduce.

^ | ⌄ • Reply • Share ›

ALSO ON **HTTPS://WWW.EDUREKA.CO/BLOG/**

What Is Microservices – Introduction To Microservice Architecture

1 comment • 6 months ago

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

Top 75 Talend Interview Questions and Answers for 2018

2 comments • 5 months ago

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

Subscribe
to our newsletter

Enter your Email Address

SUBSCRIBE

Related Blogs

(https://www.edureka.co/blog/interview-questions/hadoop-interview-questions-hdfs-2/)

Top Hadoop Interview Questions To Prepare In 2018 – HDFS (https://www.edureka.co/blog/interview-questions/hadoop-interview-questions-hdfs-2/)

(https://www.edureka.co/blog/apache-hadoop-hdfs-architecture/)

Apache Hadoop HDFS Architecture (https://www.edureka.co/blog/apache-hadoop-hdfs-architecture/)

(https://www.edureka.co/blog/10-reasons-why-big-data-analytics-is-the-best-career-move)

10 Reasons Why Big Data Analytics is the Best Career... (https://www.edureka.co/blog/10-reasons-why-big-data-analytics-is-the-best-career-move)

(https://www.edureka.co/blog/5-reasons-to-learn-hadoop)

5 Reasons to Learn Hadoop (https://www.edureka.co/blog/5-reasons-to-learn-hadoop)

Edureka

About us (https://www.edureka.co/about-us)

Blog (https://www.edureka.co/blog/)

Work with us

Become an Instructor (https://www.edureka.co/instructors/add)

Follow us on

f 🐦 in ▶

Learn on the GO!

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

News & Media
(https://www.edureka.co/media)
Contact us
(https://www.edureka.co/contact-us)
Careers
(https://www.edureka.co/careers)

Reviews
(https://www.edureka.co/reviews)
Terms & conditions
(https://www.edureka.co/terms-and-conditions)
Privacy policy
(https://www.edureka.co/privacy-policy)
Sitemap
(https://www.edureka.co/sitemap)

Hire from Edureka
(https://www.edureka.co/hire-from-edureka)

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

edureka!
(https://www.edureka.co)