

# International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: [www.ijarcsms.com](http://www.ijarcsms.com)

## “Hadoop” An Open Source OS - An Introduction

Ansari Asif<sup>1</sup>

Lecturer

Babasaheb Gawde Institute of Technology  
Mumbai – IndiaAjit Parab<sup>2</sup>

H.O.D,

Babasaheb Gawde Institute of Technology  
Mumbai – India

**Abstract:** *Hadoop. This open source software platform managed by the Apache Software Foundation has proven to be very helpful in storing and managing vast amounts of data cheaply and efficiently.*

*But what exactly is Hadoop, and what makes it so special? Basically, it's a way of storing enormous data sets across distributed clusters of servers and then running "distributed" analysis applications in each cluster.*

### I. INTRODUCTION

Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation.

Hadoop makes it possible to run applications on systems with thousands of nodes involving thousands of terabytes. Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating uninterrupted in case of a node failure. This approach lowers the risk of catastrophic system failure, even if a significant number of nodes become inoperative.

For example, suppose you have 10 terabytes of data, and you want to process it somehow, (suppose you need to sort it). Using a single computer, this could take a *very* long time. Traditionally, a high-end super-computer with exotic hardware would be required to do this in a reasonable amount of time.

Hadoop provides a framework to process data of this size using a computing cluster made from normal, commodity hardware. There are two major components to Hadoop: the file system, which is a distributed file system that splits up large files onto multiple computers, and the MapReduce framework, which is an application framework used to process large data stored on the file system.

Of course, there are many distributed computing frameworks, but what is particularly notable about Hadoop (and Google's MapReduce) is the built-in *fault tolerance*. It is designed to run on commodity hardware, and therefore it *expects* computers to be breaking frequently. The underlying file system is highly-redundant (blocks of data are replicated across multiple computers) and the MapReduce processing framework automatically handles computer failures which occur during a processing job by reassigning the processing to another computer in the cluster.

#### ***Hadoop has two main parts:***

It has two main parts - a data processing framework and a distributed filesystem for data storage. There's more to it than that, of course, but those two components really make things go.

The distributed filesystem is that far-flung array of storage clusters noted above - i.e., the Hadoop component that holds the actual data. By default, Hadoop uses the cleverly named Hadoop Distributed File System (HDFS), although it can use other file systems as well.

HDFS is like the bucket of the Hadoop system: You dump in your data and it sits there all nice and cozy until you want to do something with it, whether that's running an analysis on it within Hadoop or capturing and exporting a set of data to another tool and performing the analysis there.

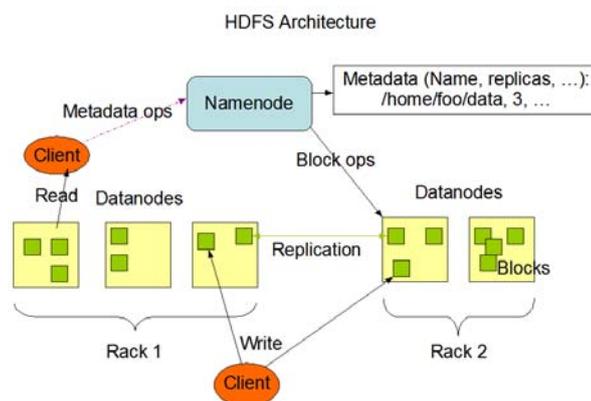
### **HDFS Architecture**

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients.

In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on.

HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories.

It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.



The NameNode and DataNode are pieces of software designed to run on commodity machines. These machines typically run a GNU/Linux operating system (OS). HDFS is built using the Java language; any machine that supports Java can run the NameNode or the DataNode software. Usage of the highly portable Java language means that HDFS can be deployed on a wide range of machines. A typical deployment has a dedicated machine that runs only the NameNode software. Each of the other machines in the cluster runs one instance of the DataNode software. The architecture does not preclude running multiple DataNodes on the same machine but in a real deployment that is rarely the case.

The existence of a single NameNode in a cluster greatly simplifies the architecture of the system. The NameNode is the arbitrator and repository for all HDFS metadata. The system is designed in such a way that user data never flows through the NameNode.

### **The Communication Protocols**

All HDFS communication protocols are layered on top of the TCP/IP protocol. A client establishes a connection to a configurable TCP port on the NameNode machine. It talks the ClientProtocol with the NameNode. The DataNodes talk to the NameNode using the DataNode Protocol. A Remote Procedure Call (RPC) abstraction wraps both the Client Protocol and the DataNode Protocol. By design, the NameNode never initiates any RPCs. Instead, it only responds to RPC requests issued by DataNodes or clients.

HDFS delivers inexpensive, reliable, and available file storage. That service alone, though, would not be enough to create the level of interest, or to drive the rate of adoption, that characterizes Hadoop over the past several years. The second major component of Hadoop is the parallel data processing system called MapReduce. MapReduce: Resource manager and Scheduler works as the given job is broken down in to tasks, then tasks are scheduled to be as close to data as possible.

Three levels of data locality:

- » Same server as data (local disk)
- » Same rack as data (rack/leaf switch)
- » Wherever there is a free slot (cross rack)

Optimized for batch processing and failure Recovery System detects laggard tasks and speculatively executes parallel tasks on the same slice of data

## II. APPLICATION OF HADOOP

The HDFS file system is not restricted to MapReduce jobs. It can be used for other applications, many of which are under development at Apache. The list includes the HBase database, the Apache Mahout Machine learning system, and the Apache Hive Data Warehouse system. Hadoop can in theory be used for any sort of work that is batch-oriented rather than real-time, that is very data-intensive, and able to work on pieces of the data in parallel. As of October 2009, commercial applications of Hadoop included:

- » Log and/or click stream analysis of various kinds
- » Marketing analytics
- » Machine learning and/or sophisticated data mining
- » Image processing
- » Processing of XML messages
- » Web crawling and/or text processing
- » General archiving, including of relational/tabular data, e.g. for compliance.

## References

1. Dean, J. and Ghemawat, S., "MapReduce: Simplified Data Processing on Large Clusters." Appeared in Proceedings of the Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004. Available online at <http://labs.google.com/papers/mapreduce.html>, March 2010.
2. Apache Nutch project, <http://lucene.apache.org/nutch/>, March 2010.
3. Apache Hadoop project, <http://hadoop.apache.org/>, March 2010.
4. Patterson, J., "The Smart Grid: Hadoop at the Tennessee Valley Authority (TVA)," Cloudera Blog, <http://www.cloudera.com/blog/2009/06/smart-grid-hadooptennessee-valley-authority-tva/>, June 2, 2009. 65 International Journal of Computer Science Research & Technology (IJCSRT) ISSN: 2321-8827 IJCSRT/IJCSRT IJCSRTV1IS040079 www.ijcsrt.org Vol. 1 Issue 4, September – 2013
5. White, T., Hadoop: The Definitive Guide. O'Reilly Media, May 2009.
6. Lin, J. and Dyer, C., Data-Intensive Text Processing with MapReduce. To be published by Morgan and Claypool. Pre-press edition available at <http://www.umiacs.umd.edu/~jimmylin/MapReducebook-20100219.pdf>, February 2010.
7. Segaran, T. and Hammerbacher, J., Beautiful Data. O'Reilly Media, July 2009.
8. The Hadoop Distributed File System: Architecture and Design
9. Yaniv Pessach (2013). Distributed Storage (Distributed Storage: Concepts, Algorithms, and Implementations ed.). Amazon
10. How 30+ enterprises are using Hadoop", in DBMS2.