# Apache Hadoop

**Apache Hadoop** is an open-source software framework for storage and large-scale processing of data-sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users.[2] It is licensed under the Apache License 2.0.

The Apache Hadoop framework is composed of the following modules:

- *Hadoop Common* – contains libraries and utilities needed by other Hadoop modules.
- *Hadoop Distributed File System (HDFS)* – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
- *Hadoop YARN* – a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
- *Hadoop MapReduce* – a programming model for large scale data processing.

All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines, or racks of machines) are common and thus should be automatically handled in software by the framework. Apache Hadoop's MapReduce and HDFS components originally derived respectively from Google's MapReduce and Google File System (GFS) papers.

YARN stands for "Yet Another Resource Negotiator" and was added later as part of Hadoop 2.0. YARN takes the resource management capabilities that were in MapReduce and packages them so they can be used by new engines. This also streamlines MapReduce to do what it does best, process data. With YARN, you can now run multiple applications in Hadoop, all sharing a common resource management. As of September, 2014, YARN manages only CPU (number of cores) and memory,[3] but management of other resources such as disk, network and GPU is planned for the future.[4]

Beyond HDFS, YARN, and MapReduce, the entire Apache Hadoop "platform" is now commonly considered to consist of a number of related projects as well – Apache Pig, Apache Hive, Apache HBase, Apache Spark, and others.[5]

For the end-users, though MapReduce Java code is common, any programming language can be used with "Hadoop Streaming" to implement the "map" and "reduce" parts of the user's program.[6] Apache Pig, Apache Hive, Apache Spark among other related projects expose higher level user interfaces like Pig Latin and a SQL variant respectively. The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command line utilities written as shell-scripts.

Apache Hadoop is a registered trademark of the Apache Software Foundation.

## 1 History

Hadoop was created by Doug Cutting and Mike Cafarella[7] in 2005. Cutting, who was working at Yahoo! at the time,[8] named it after his son's toy elephant.[9] It was originally developed to support distribution for the Nutch search engine project.[10]

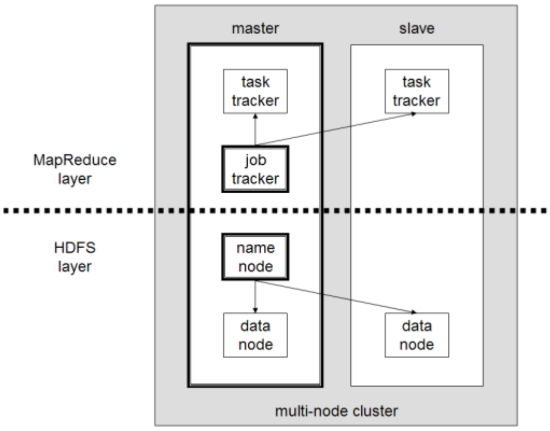
## 2 Architecture

See also: Hadoop Distributed File System, Apache HBase and MapReduce

Hadoop consists of the *Hadoop Common* package, which provides filesystem and OS level abstractions, a MapReduce engine (either MapReduce/MR1 or YARN/MR2)[11] and the Hadoop Distributed File System (HDFS). The Hadoop Common package contains the necessary Java ARchive (JAR) files and scripts needed to start Hadoop. The package also provides source code, documentation, and a contribution section that includes projects from the Hadoop Community.

For effective scheduling of work, every Hadoop-compatible file system should provide location awareness: the name of the rack (more precisely, of the network switch) where a worker node is. Hadoop applications can use this information to run work on the node where the data is, and, failing that, on the same rack/switch, reducing backbone traffic. HDFS uses this method when replicating data to try to keep different copies of the data on different racks. The goal is to reduce the impact of a rack power outage or switch failure, so that even if these events occur, the data may still be readable.[12]

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of



*A multi-node Hadoop cluster*

a JobTracker, TaskTracker, NameNode and DataNode. A slave or *worker node* acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications.[13] Hadoop requires Java Runtime Environment (JRE) 1.6 or higher. The standard startup and shutdown scripts require that Secure Shell (ssh) be set up between nodes in the cluster.[14]

In a larger cluster, the HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary NameNode that can generate snapshots of the namenode's memory structures, thus preventing file-system corruption and reducing loss of data. Similarly, a standalone JobTracker server can manage job scheduling. In clusters where the Hadoop MapReduce engine is deployed against an alternate file system, the NameNode, secondary NameNode, and DataNode architecture of HDFS are replaced by the file-system-specific equivalents.

## 2.1 File system

### 2.1.1 Hadoop distributed file system

The **Hadoop distributed file system** (**HDFS**) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. A Hadoop cluster has nominally a single namenode plus a cluster of datanodes, although redundancy options are available for the namenode due to its criticality. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses TCP/IP sockets for communication. Clients use remote procedure call (RPC) to communicate between each other.

HDFS stores large files (typically in the range of gigabytes to terabytes[15]) across multiple machines. It achieves reliability by replicating the data across multiple hosts, and hence theoretically does not require RAID storage on hosts (but to increase I/O performance some RAID configurations are still useful). With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack. Data nodes can talk to each other to rebalance data, to move copies around, and to keep the replication of data high. HDFS is not fully POSIX-compliant, because the requirements for a POSIX file-system differ from the target goals for a Hadoop application. The tradeoff of not having a fully POSIX-compliant file-system is increased performance for data throughput and support for non-POSIX operations such as Append.[16]

HDFS added the high-availability capabilities, as announced for release 2.0 in May 2012,[17] letting the main metadata server (the NameNode) fail over manually to a backup. The project has also started developing automatic fail-over.

The HDFS file system includes a so-called *secondary namenode,* a misleading name that some might incorrectly interpreted as a backup namenode for when the primary namenode goes offline. In fact, the secondary namenode regularly connects with the primary namenode and builds snapshots of the primary namenode's directory information, which the system then saves to local or remote directories. These checkpointed images can be used to restart a failed primary namenode without having to replay the entire journal of file-system actions, then to edit the log to create an up-to-date directory structure. Because the namenode is the single point for storage and management of metadata, it can become a bottleneck for supporting a huge number of files, especially a large number of small files. HDFS Federation, a new addition, aims to tackle this problem to a certain extent by allowing multiple namespaces served by separate namenodes.

An advantage of using HDFS is data awareness between the job tracker and task tracker. The job tracker schedules map or reduce jobs to task trackers with an awareness of the data location. For example: if node A contains data (x,y,z) and node B contains data (a,b,c), the job tracker schedules node B to perform map or reduce tasks on (a,b,c) and node A would be scheduled to perform map or reduce tasks on (x,y,z). This reduces the amount of traffic that goes over the network and prevents unnecessary data transfer. When Hadoop is used with other file systems, this advantage is not always available. This can have a significant impact on job-completion times, which has been demonstrated when running data-intensive jobs.[18]

HDFS was designed for mostly immutable files[16] and may not be suitable for systems requiring concurrent write-operations.

HDFS can be mounted directly with a Filesystem in Userspace (FUSE) virtual file system on Linux and some other Unix systems.

File access can be achieved through the native Java API, the Thrift API to generate a client in the language of the users' choosing (C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, Smalltalk, and OCaml), the command-line interface, browsed through the HDFS-UI webapp over HTTP, or via 3rd-party network client libraries.[19]

### 2.1.2 Other file systems

Hadoop works directly with any distributed file system that can be mounted by the underlying operating system simply by using a file:// URL; however, this comes at a price: the loss of locality. To reduce network traffic, Hadoop needs to know which servers are closest to the data; this is information that Hadoop-specific file system bridges can provide.

In May 2011, the list of supported file systems bundled with Apache Hadoop were:

- HDFS: Hadoop's own rack-aware file system.[20] This is designed to scale to tens of petabytes of storage and runs on top of the file systems of the underlying operating systems.
- FTP File system: this stores all its data on remotely accessible FTP servers.
- Amazon S3 file system. This is targeted at clusters hosted on the Amazon Elastic Compute Cloud server-on-demand infrastructure. There is no rack-awareness in this file system, as it is all remote.
- Windows Azure Storage Blobs (WASB) file system. WASB, an extension on top of HDFS, allows distributions of Hadoop to access data in Azure blob stores without moving the data permanently into the cluster.

A number of third-party file system bridges have also been written, none of which are currently in Hadoop distributions. However, some commercial distributions of Hadoop ship with an alternative filesystem as the default, -specifically IBM and MapR.

- In 2009 IBM discussed running Hadoop over the IBM General Parallel File System.[21] The source code was published in October 2009.[22]
- In April 2010, Parascale published the source code to run Hadoop against the Parascale file system.[23]
- In April 2010, Appistry released a Hadoop file system driver for use with its own CloudIQ Storage product.[24]
- In June 2010, HP discussed a location-aware IBRIX Fusion file system driver.[25]
- In May 2011, MapR Technologies, Inc. announced the availability of an alternative file system for Hadoop, which replaced the HDFS file system with a full random-access read/write file system.

## 2.2 JobTracker and TaskTracker: the MapReduce engine

Main article: MapReduce

Above the file systems comes the MapReduce engine, which consists of one *JobTracker*, to which client applications submit MapReduce jobs. The JobTracker pushes work out to available *TaskTracker* nodes in the cluster, striving to keep the work as close to the data as possible. With a rack-aware file system, the JobTracker knows which node contains the data, and which other machines are nearby. If the work cannot be hosted on the actual node where the data resides, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network. If a TaskTracker fails or times out, that part of the job is rescheduled. The TaskTracker on each node spawns off a separate Java Virtual Machine process to prevent the TaskTracker itself from failing if the running job crashes the JVM. A heartbeat is sent from the TaskTracker to the JobTracker every few minutes to check its status. The Job Tracker and TaskTracker status and information is exposed by Jetty and can be viewed from a web browser.

If the JobTracker failed on Hadoop 0.20 or earlier, all ongoing work was lost. Hadoop version 0.21 added some checkpointing to this process; the JobTracker records what it is up to in the file system. When a JobTracker starts up, it looks for any such data, so that it can restart work from where it left off.

Known limitations of this approach are:

- The allocation of work to TaskTrackers is very simple. Every TaskTracker has a number of available *slots* (such as “4 slots”). Every active map or reduce task takes up one slot. The Job Tracker allocates work to the tracker nearest to the data with an available slot. There is no consideration of the current system load of the allocated machine, and hence its actual availability.
- If one TaskTracker is very slow, it can delay the entire MapReduce job – especially towards the end of a job, where everything can end up waiting for the slowest task. With speculative execution enabled, however, a single task can be executed on multiple slave nodes.

### 2.2.1 Scheduling

By default Hadoop uses FIFO, and optionally 5 scheduling priorities to schedule jobs from a work queue.[26] In version 0.19 the job scheduler was refactored out of the JobTracker, while adding the ability to use an alternate scheduler (such as the *Fair scheduler* or the *Capacity scheduler*, described next).[27]

**Fair scheduler** The fair scheduler was developed by Facebook.[28] The goal of the fair scheduler is to provide fast response times for small jobs and QoS for production jobs. The fair scheduler has three basic concepts.[29]

1. Jobs are grouped into pools.
2. Each pool is assigned a guaranteed minimum share.
3. Excess capacity is split between jobs.

By default, jobs that are uncategorized go into a default pool. Pools have to specify the minimum number of map slots, reduce slots, and a limit on the number of running jobs.

**Capacity scheduler** The capacity scheduler was developed by Yahoo. The capacity scheduler supports several features that are similar to the fair scheduler.[30]

- Jobs are submitted into queues.
- Queues are allocated a fraction of the total resource capacity.
- Free resources are allocated to queues beyond their total capacity.
- Within a queue a job with a high level of priority has access to the queue's resources.

There is no preemption once a job is running.

### 2.3 Other applications

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, is very data-intensive, and benefits from parallel processing of data. It can also be used to complement a real-time system, such as lambda architecture.

As of October 2009, commercial applications of Hadoop[31] included:

- Log and/or clickstream 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

## 3 Prominent users

### 3.1 Yahoo!

On February 19, 2008, Yahoo! Inc. launched what it claimed was the world's largest Hadoop production application. The Yahoo! Search Webmap is a Hadoop application that runs on a more than 10,000 core Linux cluster and produces data that was used in every Yahoo! web search query.[32]

There are multiple Hadoop clusters at Yahoo! and no HDFS file systems or MapReduce jobs are split across multiple datacenters. Every Hadoop cluster node bootstraps the Linux image, including the Hadoop distribution. Work that the clusters perform is known to include the index calculations for the Yahoo! search engine.

On June 10, 2009, Yahoo! made the source code of the version of Hadoop it runs in production available to the public.[33] Yahoo! contributes all the work it does on Hadoop to the open-source community. The company's developers also fix bugs, provide stability improvements internally, and release this patched source code so that other users may benefit from their effort.

### 3.2 Facebook

In 2010 Facebook claimed that they had the largest Hadoop cluster in the world with 21 PB of storage.[34] On June 13, 2012 they announced the data had grown to 100 PB.[35] On November 8, 2012 they announced the data gathered in the warehouse grows by roughly half a PB per day.[36]

### 3.3 Other users

As of 2013, Hadoop adoption is widespread. For example, more than half of the Fortune 50 use Hadoop.[37]

## 4 Hadoop hosted in the Cloud

Hadoop can be deployed in a traditional onsite datacenter as well as in the cloud.[38] The cloud allows organizations to deploy Hadoop without hardware to acquire or specific setup expertise.[39] Vendors who currently have an offer for the cloud include Microsoft, Amazon, and Google.

### 4.1 Hadoop on Microsoft Azure

Azure HDInsight [40] is a service that deploys Hadoop on Microsoft Azure. HDInsight uses a Windows-based Hadoop distribution that was jointly developed with Hortonworks and allows programming extensions with .NET (in addition to Java).[40] By deploying HDInsight in the cloud, organizations can spin up the number of nodes they want and only get charged for the compute and storage that is used.[40] Hortonworks implementations can also move data from the on-premises datacenter to the cloud for backup, development/test, and bursting scenarios.[40]

### 4.2 Hadoop on Amazon EC2/S3 services

It is possible to run Hadoop on Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3).[41] As an example The New York Times used 100 Amazon EC2 instances and a Hadoop application to process 4 TB of raw image TIFF data (stored in S3) into 11 million finished PDFs in the space of 24 hours at a computation cost of about $240 (not including bandwidth).[42]

There is support for the S3 file system in Hadoop distributions, and the Hadoop team generates EC2 machine images after every release. From a pure performance perspective, Hadoop on S3/EC2 is inefficient, as the S3 file system is remote and delays returning from every write operation until the data is guaranteed not lost. This removes the locality advantages of Hadoop, which schedules work near data to save on network load.

### 4.3 Amazon Elastic MapReduce

Elastic MapReduce (EMR)[43] was introduced by Amazon in April 2009. Provisioning of the Hadoop cluster, running and terminating jobs, and handling data transfer between EC2(VM) and S3(Object Storage) are automated by Elastic MapReduce. Apache Hive, which is built on top of Hadoop for providing data warehouse services, is also offered in Elastic MapReduce.[44]

Support for using Spot Instances[45] was later added in August 2011.[46] Elastic MapReduce is fault tolerant for slave failures,[47] and it is recommended to only run the Task Instance Group on spot instances to take advantage of the lower cost while maintaining availability.[48]

## 5 Industry support of academic clusters

IBM and Google announced an initiative in 2007 to use Hadoop to support university courses in distributed computer programming.[49]

In 2008 this collaboration, the Academic Cloud Computing Initiative (ACCI), partnered with the National Science Foundation to provide grant funding to academic researchers interested in exploring large-data applications. This resulted in the creation of the Cluster Exploratory (CLuE) program.[50]

## 6 Running Hadoop in compute farm environments

Hadoop can also be used in compute farms and high-performance computing environments. Instead of setting up a dedicated Hadoop cluster, an existing compute farm can be used if the resource manager of the cluster is aware of the Hadoop jobs, and thus Hadoop jobs can be scheduled like other jobs in the cluster.

### 6.1 Condor integration

The Condor High-Throughput Computing System integration was presented at the *Condor Week* conference in 2010.[51]

## 7 Commercial support

A number of companies offer commercial implementations or support for Hadoop.[52]

### 7.1 ASF's view on the use of "Hadoop" in product names

The Apache Software Foundation has stated that only software officially released by the Apache Hadoop Project can be called *Apache Hadoop* or *Distributions of Apache Hadoop*.[53] The naming of products and derivative works from other vendors and the term "compatible" are somewhat controversial within the Hadoop developer community.[54]

## 8 Papers

Some papers influenced the birth and growth of Hadoop and big data processing. Here is a partial list:

- 2004 MapReduce: Simplified Data Processing on Large Clusters by Jeffrey Dean and Sanjay Ghemawat from Google Lab. This paper inspired Doug Cutting to develop an open-source implementation of the Map-Reduce framework. He named it Hadoop, after his son's toy elephant.
- 2005 From Databases to Dataspaces: A New Abstraction for Information Management, the authors highlight the need for storage systems to accept all data formats and to provide APIs for data access that evolve based on the storage system's understanding of the data.
- 2006 Bigtable: A Distributed Storage System for Structured Data from Google Lab.

- 2008 H-store: a high-performance, distributed main memory transaction processing system
- 2009 MAD Skills: New Analysis Practices for Big Data
- 2011 Apache Hadoop Goes Realtime at Facebook

## 9 See also

- Apache Accumulo – Secure Big Table
- Apache Bigtop - Packaging and interoperability testing of Hadoop-related projects
- Apache Cassandra – A column-oriented database that supports access from Hadoop
- Apache CouchDB is a database that uses JSON for documents, JavaScript for MapReduce queries, and regular HTTP for an API
- Apache Mahout – Machine Learning algorithms implemented on Hadoop
- Big data
- Cloud computing
- Data Intensive Computing
- Datameer Analytics Solution (DAS) – data source integration, storage, analytics engine and visualization
- Druid (open-source data store) - Provides a native indexing service for ingesting from HDFS.
- HBase – BigTable-model database
- HPCC – LexisNexis Risk Solutions High Performance Computing Cluster
- Hortonworks - Designed, developed, and built completely in the open, Hortonworks Data Platform (HDP) provides Hadoop designed to meet the needs of enterprise data processing
- Hypertable – HBase alternative
- MapReduce – Hadoop's fundamental data filtering algorithm
- Nutch – An effort to build an open source search engine based on Lucene and Hadoop, also created by Doug Cutting
- Pentaho – Open source data integration (Kettle), analytics, reporting, visualization and predictive analytics directly from Hadoop nodes
- Pivotal HD - Apache Hadoop distribution enhanced to support enterprise Big Data analytics. Industry's first native massively parallel processing (MPP) SQL database on Hadoop.
- Qubole - a cloud-based Big Data as a service developer
- RapidMiner Radoop – In-Hadoop big data analytics providing a set of algorithms for doing scalable data transformations, advanced analytics, and predictive modeling
- Sector/Sphere – Open source distributed storage and processing
- Simple Linux Utility for Resource Management
- Talend – An open source integration software

## 10 References

[1] "Hadoop Releases". Hadoop.apache.org. Retrieved 2014-06-30.

[2] "Applications and organizations using Hadoop". Wiki.apache.org. 2013-06-19. Retrieved 2013-10-17.

[3] "Resource (Apache Hadoop Main 2.5.1 API)". *apache.org*. Apache Software Foundation. 2014-09. Retrieved 2014-09-30. Check date values in: |date= (help)

[4] Murthy, Arun (2012-08-15). "Apache Hadoop YARN – Concepts and Applications". *hortonworks.com*. Hortonworks. Retrieved 2014-09-30.

[5] "Hadoop-related projects at". Hadoop.apache.org. Retrieved 2013-10-17.

[6] "[nlpatumd] Adventures with Hadoop and Perl". Mail-archive.com. 2010-05-02. Retrieved 2013-04-05.

[7] "Michael J. Cafarella". Web.eecs.umich.edu. Retrieved 2013-04-05.

[8] Hadoop creator goes to Cloudera

[9] Ashlee Vance (2009-03-17). "Hadoop, a Free Software Program, Finds Uses Beyond Search". *The New York Times*. Archived from the original on 11 February 2010. Retrieved 2010-01-20.

[10] "Hadoop contains the distributed computing platform that was formerly a part of Nutch. This includes the Hadoop Distributed Filesystem (HDFS) and an implementation of MapReduce." About Hadoop

[11] Harsh Chouraria (21 October 2012). "MR2 and YARN Briefly Explained". *cloudera.com*. Cloudera. Retrieved 23 October 2013.

[12] "HDFS User Guide". Hadoop.apache.org. Retrieved 2014-09-04.

[13] "Running Hadoop on Ubuntu Linux (Multi-Node Cluster)".

[14] "Running Hadoop on Ubuntu Linux (Single-Node Cluster)". Retrieved 6 June 2013.

[15] "HDFS Architecture". Retrieved 1 September 2013.

[16] Yaniv Pessach (2013). "Distributed Storage" (Distributed Storage: Concepts, Algorithms, and Implementations ed.). Amazon.com

[17] "Version 2.0 provides for manual failover and they are working on automatic failover:". Hadoop.apache.org. Retrieved 30 July 2013.

[18] "Improving MapReduce performance through data placement in heterogeneous Hadoop Clusters" (PDF). Eng.auburn.ed. April 2010.

[19] "Mounting HDFS". Retrieved May 2014.

[20] "HDFS Users Guide – Rack Awareness". Hadoop.apache.org. Retrieved 2013-10-17.

[21] "Cloud analytics: Do we really need to reinvent the storage stack?". IBM. June 2009.

[22] "HADOOP-6330: Integrating IBM General Parallel File System implementation of Hadoop Filesystem interface". IBM. 2009-10-23.

[23] "HADOOP-6704: add support for Parascale filesystem". Parascale. 2010-04-14.

[24] "HDFS with CloudIQ Storage". Appistry,Inc. 2010-07-06.

[25] "High Availability Hadoop". HP. 2010-06-09.

[26] job

[27] "Refactor the scheduler out of the JobTracker". *Hadoop Common*. Apache Software Foundation. Retrieved 9 June 2012.

[28] M. Tim Jones (6 December 2011). "Scheduling in Hadoop". *ibm.com*. IBM. Retrieved 20 November 2013.

[29] Hadoop Fair Scheduler Design Document

[30] Capacity Scheduler Guide

[31] October 10, 2009 (2009-10-10). ""How 30+ enterprises are using Hadoop", in DBMS2". Dbms2.com. Retrieved 2013-10-17.

[32] Yahoo! Launches World's Largest Hadoop Production Application (Hadoop and Distributed Computing at Yahoo!)

[33] "Hadoop and Distributed Computing at Yahoo!". Yahoo!. 2011-04-20. Retrieved 2013-10-17.

[34] "HDFS: Facebook has the world's largest Hadoop cluster!". Hadoopblog.blogspot.com. 2010-05-09. Retrieved 2012-05-23.

[35] "Under the Hood: Hadoop Distributed File system reliability with Namenode and Avatarnode". Facebook. Retrieved 2012-09-13.

[36] "Under the Hood: Scheduling MapReduce jobs more efficiently with Corona". Facebook. Retrieved 2012-11-09.

[37] "Altior's AltraSTAR – Hadoop Storage Accelerator and Optimizer Now Certified on CDH4 (Cloudera's Distribution Including Apache Hadoop Version 4)" (Press release). Eatontown, New Jersey: Altior Inc. 2012-12-18. Retrieved 2013-10-30.

[38] {title=What is Hadoop?"| URL=http://azure.microsoft.com/en-us/solutions/hadoop/ }

[39] "Hadoop". Azure.microsoft.com. Retrieved 2014-07-22.

[40] "HDInsight | Cloud Hadoop". Azure.microsoft.com. Retrieved 2014-07-22.

[41] Varia, Jinesh (@jinman). "Taking Massive Distributed Computing to the Common Man – Hadoop on Amazon EC2/S3". *Amazon Web Services Blog*. Amazon.com. Retrieved 9 June 2012.

[42] Gottfrid, Derek (November 1, 2007). "Self-service, Prorated Super Computing Fun!". *The New York Times*. Retrieved May 4, 2010.

[43] "AWS | Amazon Elastic MapReduce (EMR) | Hadoop MapReduce in the Cloud". Aws.amazon.com. Retrieved 2014-07-22.

[44] "Amazon Elastic MapReduce Developer Guide" (PDF). Retrieved 2013-10-17.

[45] "Amazon EC2 Spot Instances". Aws.amazon.com. Retrieved 2014-07-22.

[46] "Amazon Elastic MapReduce Now Supports Spot Instances". Amazon.com. 2011-08-18. Retrieved 2013-10-17.

[47] "Amazon Elastic MapReduce FAQs". Amazon.com. Retrieved 2013-10-17.

[48] Using Spot Instances with EMR on YouTube

[49] "Google Press Center: Google and IBM Announce University Initiative to Address Internet-Scale Computing Challenges". Google. 2007-10-08. Retrieved 2013-10-17.

[50] "NSF, Google, IBM form CLuE". Hadoopcommunity.wordpress.com. Retrieved 2013-10-17.

[51] "Condor integrated with Hadoop's Map Reduce". University of Wisconsin–Madison. 2010-04-15.

[52] "Why the Pace of Hadoop Innovation Has to Pick Up". Gigaom.com. 2011-04-25. Retrieved 2013-10-17.

[53] "Defining Hadoop". Wiki.apache.org. 2013-03-30. Retrieved 2013-10-17.

[54] "Defining Hadoop Compatibility: revisited". Mail-archives.apache.org. 2011-05-10. Retrieved 2013-10-17.

## 11 Bibliography

- Lam, Chuck (July 28, 2010). *Hadoop in Action* (1st ed.). Manning Publications. p. 325. ISBN 1-935182-19-6.
- Venner, Jason (June 22, 2009). *Pro Hadoop* (1st ed.). Apress. p. 440. ISBN 1-4302-1942-4.
- White, Tom (June 16, 2009). *Hadoop: The Definitive Guide* (1st ed.). O'Reilly Media. p. 524. ISBN 0-596-52197-9.

## 12 External links

- Official Hadoop Homepage
- Official Hadoop Wiki
- Introducing Apache Hadoop: The Modern Data Operating System — lecture given at Stanford University by Co-Founder and CTO of Cloudera, Amr Awadallah (video archive) (YouTube).

# 13 Text and image sources, contributors, and licenses

## 13.1 Text

- **Apache Hadoop** *Source:* http://en.wikipedia.org/wiki/Apache_Hadoop?oldid=630671093 *Contributors:* Nealmcb, Michael Hardy, Kwertii, Den fjättrade ankan, Toreau, AnonMoos, Rfc1394, Superm401, Aomarks, Sepreece, Khalid hassani, Neilc, Pgan002, Alexf, Beland, Jeremykemp, SamuelScarano, Wadsworth, Thorwald, Hydrox, IlyaHaykinson, Talbrech, John Vandenberg, Jonsafari, Diego Moya, YPavan, Dan100, Oleg Alexandrov, Mindmatrix, GregorB, Marudubshinki, Kesla, Josh Parris, MikeMayer, Intgr, Ronebofh, Tas50, Wavelength, Markpeak, Joebeone, SteveLoughran, NawlinWiki, LodeRunner, SColombo, Cosmotron, BenBildstein, Banana04131, JLaTondre, Vahid83, Mlibby, SmackBot, Y10k, Faisal.akeel, McGeddon, Alkini, Gilliam, Ohnoitsjamie, Chris the speller, Thumperward, Lollerskates, HughNo, Nick Levine, Frap, Parent5446, Billytkid, Derek R Bullamore, EIFY, Mwtoews, Kuru, Stennie, Uribreitman, Kompere, JHunterJ, Caiaffa, Hu12, Alexbrewer, OS2Warp, Knoche, JerkerNyberg, Raysonho, Only2sea, Jesse Viviano, Unamigo, Emocat, Mblumber, Bilbobaggginz, Hebrides, Blaisorblade, Vinoduec, Hervegirod, Vertium, Mlogic, Gioto, KMeyer, Mutt Lunker, Labongo, NapoliRoma, Yanngeffrotin, Camerojo, Jacobbbc, Straxus, Theneoindian, Orenfalkowitz, Maju wiki, Gwern, Atree, Mukesh.khandelwal, Bostonvaulter, Jackson Peebles, GeneGotimer, Metroshica, McSly, Curlybraces, LightningDragon, Tagus, AngryBear, DaoKaioshin, Owen.omalley, Mschober65, JimJJewett, Dguedo, LittleBenW, Mingfai, PokeYourHeadOff, Redragon104, EmxBot, TJRC, Yintan, Bodzasfanta, Flyer22, Jojalozzo, Free Software Knight, Svick, Capitalismojo, Denisarona, Brian Geppert, H.E. Hall, EoGuy, Hult041956, Drmies, Kotalampi, Alexbot, Jeric14, His Wikiness, TedDunning, Rulerofutumno, DumZiBoT, Ronnie6657, XLinkBot, Ost316, Marchingknight11, Dsimic, DougCutting, Addbot, Jarvisa, Mortense, Tsunanet, Blinkov, Techlady, MrOllie, Kajaro, Renatokeshet, Leonidas from XIV, Jarble, Pmj005, Luckas-bot, BaldPark, Yobot, Themfromspace, Bunnyhop11, Sumail, WikiScrubber, Jreinsch, AnomieBOT, Rubinbot, Patodirahul, Shock an awe, CeciliaPang, GrantIngersoll, Citation bot, Arodichevski, Quebec99, TaranSingh, Anna Frodesiak, FontOfSomeKnowledge, Arjant, Tabledhote, Doronve, FrescoBot, LucienBOT, W Nowicki, Sae1962, NuclearWizard, Citation bot 1, John kreisa, Rickyphyllis, Winterst, Tsasaa12, Feliciafay, Justpossible, Jim380, Blacknosugar, Cnwilliams, Trappist the monk, Swooledge, Nameispete, Lotje, LustreOne, Akim Demaille, Dskrvk, InMktgWeTrust, RjwilmsiBot, Hadoopor, Kerrick Staley, DASHBot, EmausBot, John of Reading, WikitanvirBot, Angrytoast, Dewritech, GoingBatty, Peaceray, Snorgway, ZéroBot, John Cline, Bongoramsey, Josve05a, Liorrokach, Ebrambot, Captain Screebo, Paris.butterfield, Wikfr, Donner60, Sheilakinsella, Paulboal, Fun indra, ClueBot NG, RickFarnell, Tabletrack53, Lord.Quackstar, AgniKalpa, Nmilford, Kasirbot, Zak.estrada, Widr, CasualVisitor, Fvillanustre, Helpful Pixie Bot, Ravi.panditsadreena, HMSSolent, Dujunping, Datastage, BG19bot, Krenair, BendelacBOT, Jan Spousta, Serent, Jdcourco, Mjf19, Chafe66, Michaelmalak, Jordanzhang, Compfreak7, Craigacp, Wangchuck50, Samataha, Dietcoke3.14, BattyBot, Mg acom, Mediran, Khazar2, JYBot, IjonTichyIjonTichy, Research186, Davidogm, GriffinGrl, Kephir, Makecat-bot, Cerabot, Talkietoaster-nc, MartinMichlmayr, Nafaabout, CJGarner, 900mill, Nullwriter, Mahbubur-r-aaman, Epicgenius, Lsmll, LucienK, Jain.anilk, Mand Beckett, Windmost, Gsoverby, Bitobor, Adam kawa 85, Michaeldweir, Menelaosc, ScotXW, AGraziani, Jppcap, Rdashore37, Nyashinski, Tzolov, Vieque, Techgirl007, Textractor, Cindygross, Drummerwolli32, Bigdatahadoopjaipur, Amar.chandole, Iqmc and Anonymous: 394

## 13.2 Images

- **File:ASF-logo.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/c/cd/ASF-logo.svg *License:* ? *Contributors:* http://www.apache.org/ *Original artist:* Apache Software Foundation (ASF)
- **File:Ambox_content.png** *Source:* http://upload.wikimedia.org/wikipedia/en/f/f4/Ambox_content.png *License:* ? *Contributors:*

  Derived from Image:Information icon.svg *Original artist:*

  El T (original icon); David Levy (modified design); Penubag (modified color)
- **File:Commons-logo.svg** *Source:* http://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg *License:* ? *Contributors:* ? *Original artist:* ?
- **File:Edit-clear.svg** *Source:* http://upload.wikimedia.org/wikipedia/en/f/f2/Edit-clear.svg *License:* ? *Contributors:* The *Tango! Desktop Project*. *Original artist:*

  The people from the Tango! project. And according to the meta-data in the file, specifically: “Andreas Nilsson, and Jakub Steiner (although minimally).”
- **File:Folder_Hexagonal_Icon.svg** *Source:* http://upload.wikimedia.org/wikipedia/en/4/48/Folder_Hexagonal_Icon.svg *License:* ? *Contributors:* ? *Original artist:* ?
- **File:Free_Software_Portal_Logo.svg** *Source:* http://upload.wikimedia.org/wikipedia/commons/3/31/Free_and_open-source_software_logo_%282009%29.svg *License:* Public domain *Contributors:*
- FOSS Logo.svg *Original artist:* FOSS Logo.svg: ViperSnake151
- **File:Hadoop_1.png** *Source:* http://upload.wikimedia.org/wikipedia/en/2/2b/Hadoop_1.png *License:* ? *Contributors:* ? *Original artist:* ?

## 13.3 Content license