Apache Hadoop

From Wikipedia, the free encyclopedia

Apache Hadoop (pronunciation: /həˈduːp/) is an open-source software framework used for distributed storage and processing of very large data sets. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are a common occurrence and should be automatically handled by the framework.[2]

Apache Hadoop

Developer(s)	Apache Software Foundation
Initial release	December 10, 2011[1]
Stable release	2.7.3 / August 25, 2016[2]
Repository	git-wip-us.apache.org/repos/asf/hadoop.git (https://git-wip-us.apache.org/repos/asf/hadoop.git)
Development status	Active
Written in	Java
Operating system	Cross-platform
Type	Distributed file system
License	Apache License 2.0
Website	hadoop.apache.org (http://hadoop.apache.org)

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality[3] – nodes manipulating the data they have access to – to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.[4]

The base 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 computing resources in clusters and using them for scheduling of users' applications;[5][6] and
- *Hadoop MapReduce* – an implementation of the MapReduce programming model for large scale data processing.

The term *Hadoop* has come to refer not just to the base modules above, but also to the *ecosystem*,[7] or collection of additional software packages that can be installed on top of or alongside Hadoop, such as Apache Pig, Apache Hive, Apache HBase, Apache Phoenix, Apache Spark, Apache ZooKeeper, Cloudera Impala, Apache Flume, Apache Sqoop, Apache Oozie, Apache Storm.[8]

Apache Hadoop's MapReduce and HDFS components were inspired by Google papers on their MapReduce and Google File System.[9]

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. 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.[10] Other projects in the Hadoop ecosystem expose richer user interfaces.

Contents

History

The genesis of Hadoop came from the Google File System paper[11] that was published in October 2003. This paper spawned another research paper from Google – MapReduce: Simplified Data Processing on Large Clusters.[12] Development started on the Apache Nutch project, but was moved to the new Hadoop subproject in January 2006.[13] Doug Cutting, who was working at Yahoo! at the time,[14] named it after his son's toy elephant.[15] The initial code that was factored out of Nutch consisted of 5k lines of code for NDFS and 6k lines of code for MapReduce.

The first committer added to the Hadoop project was Owen O’Malley in March 2006.[16] Hadoop 0.1.0 was released in April 2006[17] and continues to evolve by the many contributors[18] to the Apache Hadoop project.

Timeline

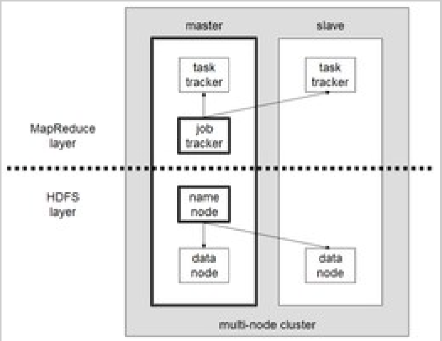
Year	Month	Event	Ref.
2003	October	Google File System paper released	[19]
2004	December	MapReduce: Simplified Data Processing on Large Clusters	[20]
2006	January	Hadoop subproject created with mailing lists, jira, and wiki	[21]
2006	January	Hadoop is born from Nutch 197	[22]
2006	February	NDFS+ MapReduce moved out of Apache Nutch to create Hadoop	[23]
2006	February	Owen Omalley's first patch goes into Hadoop	[24]
2006	February	Hadoop is named after Cutting's son's yellow plush toy	[25]
2006	April	Hadoop 0.1.0 released	[26]
2006	April	Hadoop sorts 1.8 TB on 188 nodes in 47.9 hours	[23]
2006	May	Yahoo deploys 300 machine Hadoop cluster	[23]
2006	October	Yahoo Hadoop cluster reaches 600 machines	[23]
2007	April	Yahoo runs two clusters of 1,000 machines	[23]
2007	June	Only three companies on "Powered by Hadoop Page"	[27]
2007	October	First release of Hadoop that includes HBase	[28]
2007	October	Yahoo Labs creates Pig, and donates it to the ASF	[29]
2008	January	YARN JIRA opened	Yarn Jira (Mapreduce 279)
2008	January	20 companies on "Powered by Hadoop Page"	[27]
2008	February	Yahoo moves its web index onto Hadoop	[30]
2008	February	Yahoo! production search index generated by a 10,000-core Hadoop cluster	[23]
2008	March	First Hadoop Summit	[31]
2008	April	Hadoop world record fastest system to sort a terabyte of data. Running on a 910-node cluster, Hadoop sorted one terabyte in 209 seconds	[23]
2008	May	Hadoop wins TeraByte Sort (World Record sortbenchmark.org)	[32]
2008	July	Hadoop wins Terabyte Sort Benchmark	[33]
2008	October	Loading 10 TB/day in Yahoo clusters	[23]
2008	October	Cloudera, Hadoop distributor is founded	[34]
2008	November	Google MapReduce implementation sorted one terabyte in 68 seconds	[23]
2009	March	Yahoo runs 17 clusters with 24,000 machines	[23]
2009	April	Hadoop sorts a petabyte	[35]
2009	May	Yahoo! used Hadoop to sort one terabyte in 62 seconds	[23]
2009	June	Second Hadoop Summit	[36]
2009	July	Hadoop Core is renamed Hadoop Common	[37]
2009	July	MapR, Hadoop distributor founded	[38]
2009	July	HDFS now a separate subproject	[37]
2009	July	MapReduce now a separate subproject	[37]
2010	January	Kerberos support added to Hadoop	[39]
2010	May	Apache HBase Graduates	[40]

2010	June	Third Hadoop Summit	[41]
2010	June	Yahoo 4,000 nodes/70 petabytes	[42]
2010	June	Facebook 2,300 clusters/40 petabytes	[42]
2010	September	Apache Hive Graduates	[43]
2010	September	Apache Pig Graduates	[44]
2011	January	Apache Zookeeper Graduates	[45]
2011	January	Facebook, LinkedIn, eBay and IBM collectively contribute 200,000 lines of code	[46]
2011	March	Apache Hadoop takes top prize at Media Guardian Innovation Awards	[47]
2011	June	Rob Beardon and Eric Badleschieler spin out Hortonworks out of Yahoo.	[48]
2011	June	Yahoo has 42K Hadoop nodes and hundreds of petabytes of storage	[48]
2011	June	Third Annual Hadoop Summit (1,700 attendees)	[49]
2011	October	Debate over which company had contributed more to Hadoop.	[46]
2012	January	Hadoop community moves to separate from MapReduce and replace with YARN	[25]
2012	June	San Jose Hadoop Summit (2,100 attendees)	[50]
2012	November	Apache Hadoop 1.0 Available	[37]
2013	March	Hadoop Summit – Amsterdam (500 attendees)	[51]
2013	March	YARN deployed in production at Yahoo	[52]
2013	June	San Jose Hadoop Summit (2,700 attendees)	[53]
2013	October	Apache Hadoop 2.2 Available	[37]
2014	February	Apache Hadoop 2.3 Available	[37]
2014	February	Apache Spark top Level Apache Project	[54]
2014	April	Hadoop summit Amsterdam (750 attendees)	[55]
2014	June	Apache Hadoop 2.4 Available	[37]
2014	June	San Jose Hadoop Summit (3,200 attendees)	[56]
2014	August	Apache Hadoop 2.5 Available	[37]
2014	November	Apache Hadoop 2.6 Available	[37]
2015	April	Hadoop Summit Europe	[57]
2015	June	Apache Hadoop 2.7 Available	[37]

Architecture

Hadoop consists of the *Hadoop Common* package, which provides file system and OS level abstractions, a MapReduce engine (either MapReduce/MR1 or YARN/MR2)[58] and the Hadoop Distributed File System (HDFS). The Hadoop Common package contains the necessary Java ARchive (JAR) files and scripts needed to start Hadoop.

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 execute code on the node where the data is, and, failing that, on the same rack/switch to reduce backbone traffic. HDFS uses this method when replicating data for data redundancy across multiple racks. This approach reduces the impact of a rack power outage or switch failure; if one of these hardware failures occurs, the data will remain available.[59]



A multi-node Hadoop cluster

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a Job Tracker, Task Tracker, 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.[60]

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.[61]

In a larger cluster, HDFS nodes are 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, thereby preventing file-system corruption and loss of data. Similarly, a standalone JobTracker server can manage job scheduling across nodes. When Hadoop MapReduce is used with an alternate file system, the NameNode, secondary NameNode, and DataNode architecture of HDFS are replaced by the file-system-specific equivalents.

File systems

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. Some consider HDFS to instead be a data store due to its lack of POSIX compliance and inability to be mounted,[62] but it does provide shell commands and Java API methods that are similar to other file systems.[63] 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[64]) 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 trade-off of not having a fully POSIX-compliant file-system is increased performance for data throughput and support for non-POSIX operations such as Append.[65]

HDFS added the high-availability capabilities, as announced for release 2.0 in May 2012,[66] 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 interpret 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. Moreover, there are some issues in HDFS, namely, small file issue, scalability problem, Single Point of Failure (SPoF), and bottleneck in huge metadata request. 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.[67]

HDFS was designed for mostly immutable files[65] 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 application programming interface (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 Web application (webapp) over HTTP, or via 3rd-party network client libraries.[68]

HDFS is designed for portability across various hardware platforms and compatibility with a variety of underlying operating systems. The HDFS design introduces portability limitations that result in some performance bottlenecks, since the Java implementation can't use features that are exclusive to the platform on which HDFS is running.[69] Due to its widespread integration into enterprise-level infrastructures, monitoring HDFS performance at scale has become an increasingly important issue. Monitoring end-to-end performance requires tracking metrics from datanodes, namenodes, and the underlying operating system.[70] There are currently several monitoring platforms to track HDFS performance, including HortonWorks, Cloudera, and Datadog.

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.[71] 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 (Simple Storage Service) 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) (http://azure.microsoft.com/en-us/documentation/articles/hdinsight-use-blob-storage) 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.[72] The source code was published in October 2009.[73]
- In April 2010, Parascale published the source code to run Hadoop against the Parascale file system.[74]
- In April 2010, Appistry released a Hadoop file system driver for use with its own CloudIQ Storage product.[75]
- In June 2010, HP discussed a location-aware IBRIX Fusion file system driver.[76]
- In May 2011, MapR Technologies, Inc. announced the availability of an alternative file system for Hadoop, MapR FS, which replaced the HDFS file system with a full random-access read/write file system.

JobTracker and TaskTracker: the MapReduce engine

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 a separate Java Virtual Machine process to prevent the TaskTracker itself from failing if the running job crashes its 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.

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.

Scheduling

By default Hadoop uses FIFO scheduling, and optionally 5 scheduling priorities to schedule jobs from a work queue.[77] 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).[78]

Fair scheduler

The fair scheduler was developed by Facebook.[79] 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.[80]

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.[81]

- 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.

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, Apache Storm, Flink and Spark Streaming.[82]

As of October 2009, commercial applications of Hadoop[83] 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

Prominent users

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 Linux cluster with more than 10,000 cores and produced data that was used in every Yahoo! web search query.[84] 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. In June 2009, Yahoo! made the source code of the Hadoop version it runs available to the public via the open-source community.[85]

In 2010, Facebook claimed that they had the largest Hadoop cluster in the world with 21 PB of storage.[86] In June 2012, they announced the data had grown to 100 PB[87] and later that year they announced that the data was growing by roughly half a PB per day.[88]

As of 2013, Hadoop adoption had become widespread: more than half of the Fortune 50 used Hadoop.[89]

Hadoop hosting in the cloud

Hadoop can be deployed in a traditional onsite datacenter as well as in the cloud.[90] The cloud allows organizations to deploy Hadoop without hardware to acquire or specific setup expertise.[91] Vendors who currently have an offer for the cloud include Microsoft, Amazon, IBM,[92] Google and Oracle.[93]

On Microsoft Azure

Azure HDInsight[94] is a service that deploys Hadoop on Microsoft Azure. HDInsight uses Hortonworks HDP and was jointly developed for HDI with Hortonworks. HDI allows programming extensions with .NET (in addition to Java). HDInsight also supports creation of Hadoop clusters using Linux with Ubuntu.[94] 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.[94] Hortonworks implementations can also move data from the on-premises datacenter to the cloud for backup, development/test, and bursting scenarios.[94] It is also possible to run Cloudera or Hortonworks Hadoop clusters on Azure Virtual Machines.

On Amazon EC2/S3 services

It is possible to run Hadoop on Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3).[95] 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).[96]

There is support for the S3 object store in the Apache Hadoop releases, though this is below what one expects from a traditional POSIX filesystem. Specifically, operations such as rename() and delete() on directories are not atomic, and can take time proportional to the number of entries and the amount of data in them.

Amazon Elastic MapReduce

Elastic MapReduce (EMR)[97] was introduced by Amazon.com 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.[98]

Support for using Spot Instances[99] was later added in August 2011.[100] Elastic MapReduce is fault-tolerant for slave failures,[101] and it is recommended to only run the Task Instance Group on spot instances to take advantage of the lower cost while maintaining availability.[102]

On CenturyLink Cloud (CLC)

CenturyLink Cloud[103] offers Hadoop via both a managed and un-managed model via their Hadoop[104] offering. CLC also offers customers several managed Cloudera Blueprints, the newest managed service in the CenturyLink Cloud big data Blueprints portfolio, which also includes Cassandra and MongoDB solutions.[105]

Google Cloud Platform

There are multiple ways to run the Hadoop ecosystem on Google Cloud Platform ranging from self-managed to Google-managed.[106]

- Google Cloud Dataproc – a managed Spark and Hadoop service[107]
- command line tools (bdutil) (https://cloud.google.com/hadoop/bdutil) – a collection of shell scripts to manually create and manage Spark and Hadoop clusters[108]
- third party Hadoop distributions:-
    - Cloudera – using the Cloudera Director Plugin for Google Cloud Platform[109]
    - Hortonworks – using bdutil support for Hortonworks HDP[110]
    - MapR – using bdutil support for MapR[111]

Google also offers connectors for using other Google Cloud Platform products with Hadoop, such as a Google Cloud Storage connector (https://www.mapr.com/resources/mapr-google-cloud-platform) for using Google Cloud Storage and a Google BigQuery connector (https://cloud.google.com/hadoop/bigquery-connector) for using Google BigQuery.

Commercial support

A number of companies offer commercial implementations or support for Hadoop.[112]

Branding

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*.[113] The naming of products and derivative works from other vendors and the term "compatible" are somewhat controversial within the Hadoop developer community.[114]

Papers

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

- Jeffrey Dean, Sanjay Ghemawat (2004) MapReduce: Simplified Data Processing on Large Clusters (https://www.usenix.org/legacy/publications/library/proceedings/osdi04/tech/full_papers/dean/dean_html/index.html), Google. 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.
- Michael Franklin, Alon Halevy, David Maier (2005) From Databases to Dataspaces: A New Abstraction for Information Management (http://www.eecs.berkeley.edu/~franklin/Papers/dataspaceSR.pdf). 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.
- Fay Chang et al. (2006) Bigtable: A Distributed Storage System for Structured Data (http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/bigtable-osdi06.pdf), Google.
- Robert Kallman et al. (2008) H-store: a high-performance, distributed main memory transaction processing system (http://www.vldb.org/pvldb/1/1454211.pdf)

See also

- Apache Accumulo – Secure BigTable[115]
- 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
- Big data
- Cloud computing
- Data Intensive Computing
- HPCC – LexisNexis Risk Solutions High Performance Computing Cluster
- Hypertable – HBase alternative
- Sector/Sphere – Open source distributed storage and processing
- Simple Linux Utility for Resource Management

References

1. "Hadoop Releases". *apache.org*. Apache Software Foundation. Retrieved 2014-12-06.
2. "Welcome to Apache Hadoop!". *hadoop.apache.org*. Retrieved 2016-08-25.
3. "What is the Hadoop Distributed File System (HDFS)?". *ibm.com*. IBM. Retrieved 2014-10-30.
4. Malak, Michael (2014-09-19). "Data Locality: HPC vs. Hadoop vs. Spark". *datascienceassn.org*. Data Science Association. Retrieved 2014-10-30.
5. "Resource (Apache Hadoop Main 2.5.1 API)". *apache.org*. Apache Software Foundation. 2014-09-12. Retrieved 2014-09-30.
6. Murthy, Arun (2012-08-15). "Apache Hadoop YARN – Concepts and Applications". *hortonworks.com*. Hortonworks. Retrieved 2014-09-30.
7. "Continuuity Raises $10 Million Series A Round to Ignite Big Data Application Development Within the Hadoop Ecosystem". *finance.yahoo.com*. Marketwired. 2012-11-14. Retrieved 2014-10-30.
8. "Hadoop-related projects at". Hadoop.apache.org. Retrieved 2013-10-17.
9. *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons. 2014-12-19. p. 300. ISBN 9781118876220. Retrieved 2015-01-29.
10. "[nlpatumd] Adventures with Hadoop and Perl". Mail-archive.com. 2010-05-02. Retrieved 2013-04-05.
11. Ghemawat, Sanjay; Gobioff, Howard; Leung, Shun-Tak. "The Google File System".
12. Dean, Jeffrey; Ghemawat, Sanjay. "MapReduce: Simplified Data Processing on Large Clusters".
13. Cutting, Doug (28 Jan 2006). "new mailing lists request: hadoop". *issues.apache.org*. "The Lucene PMC has voted to split part of Nutch into a new sub-project named Hadoop"
14. Intellipaat. "Hadoop Creator goes to Cloudera". *Intellipaat Blog*. Retrieved 2 February 2016.
15. Vance, Ashlee (2009-03-17). "Hadoop, a Free Software Program, Finds Uses Beyond Search". *The New York Times*. Archived from the original on August 30, 2011. Retrieved 2010-01-20.
16. Cutting, Doug (30 March 2006). "[RESULT] VOTE: add Owen O'Malley as Hadoop committer". *hadoop-common-dev* (Mailing list).
17. "archive.apache.org".

18. "Apache Hadoop Project Members".
19. "Google Research Publication: The Google File System". Retrieved 2016-03-09.
20. "Google Research Publication: MapReduce". Retrieved 2016-03-09.
21. "[INFRA-700] new mailing lists request: hadoop - ASF JIRA". Retrieved 2016-03-09.
22. "[HADOOP-1] initial import of code from Nutch - ASF JIRA". Retrieved 2016-03-09.
23. White, Tom (2012). *Hadoop: The Definitive Guide* (3rd ed.). O'Reilly. ISBN 9781449328917.
24. "[NUTCH-197] NullPointerException in TaskRunner if application jar does not have "lib" directory - ASF JIRA". Retrieved 2016-03-09.
25. "From Spiders to Elephants: The History of Hadoop". Retrieved 2016-03-09.
26. "Index of /dist/hadoop/core". Retrieved 2016-03-09.
27. "Hadoop Summit 2009". Retrieved 2016-03-09.
28. "Apache Hadoop Releases". Retrieved 2016-03-09.
29. Gates, Alan (2011). *Programming Pig*. O'Reilly. p. 10. ISBN 978-1-4493-0264-1.
30. "Yahoo! Launches World's Largest Hadoop Production Application". *hadoopnew – Yahoo*. Retrieved 2016-03-09.
31. "RE: Hadoop summit / workshop at Yahoo!". Retrieved 2016-03-09.
32. http://sortbenchmark.org/YahooHadoop.pdf
33. "Apache Hadoop Wins Terabyte Sort Benchmark". *hadoopnew – Yahoo*. Retrieved 2016-03-09.
34. "Cloudera". Retrieved 2016-03-09.
35. http://sortbenchmark.org/Yahoo2009.pdf
36. http://www.mollynix.com/images_content/01commdes/hadoopschedulepdf.pdf
37. "Welcome to Apache™ Hadoop®!". Retrieved 2016-03-09.
38. "MapR Technologies". Retrieved 2016-03-09.
39. "Yahoo! Updates from Hadoop Summit 2010". Think Big Analytics. Retrieved April 25, 2016. "Baldeschwieler announced that Yahoo has released a beta test of Hadoop Security, which uses Kerberos for authentication and allows colocation of business sensitive data within the same cluster."
40. "Apache HBase – Apache HBase™ Home". Retrieved 2016-03-09.
41. "Hadoop Summit 2010 – Agenda is available!". *hadoopnew – Yahoo*. Retrieved 2016-03-09.
42. "Hadoop Summit 2010". Retrieved 2016-03-09.
43. "Apache Hive TM". Retrieved 2016-03-09.
44. "Welcome to Apache Pig!". Retrieved 2016-03-09.
45. "Apache ZooKeeper - Home". Retrieved 2016-03-09.
46. "Reality Check: Contributions to Apache Hadoop — Hortonworks". Retrieved 2016-03-09.
47. "Apache Hadoop takes top prize at Media Guardian Innovation Awards". *The Guardian*. Retrieved 2016-03-09.
48. Harris, Derrick. "The history of Hadoop: From 4 nodes to the future of data". Gigaom. Retrieved 2016-03-09.
49. "Hadoop Summit 2011: June 29th, Santa Clara Convention Center". *hadoopnew – Yahoo*. Retrieved 2016-03-09.
50. "Fifth Annual Hadoop Summit 2012 Kicks Off with Record Attendance - Hortonworks". Retrieved 2016-03-09.
51. "Hadoop Summit 2013 Amsterdam - It's A Wrap! - Hortonworks". Retrieved 2016-03-09.
52. "Hadoop at Yahoo!: More Than Ever Before". Retrieved 2016-03-09.
53. "Hadoop Summit North America 2013 Draws Record Ecosystem Support". *Business Wire*. Retrieved 2016-03-09.
54. "The Apache Software Foundation Announces Apache™ Spark™ as a Top-Level Project : The Apache Software Foundation Blog". Retrieved 2016-03-09.
55. "Loved Hadoop Summit Europe 2014 - Hope you did too! - SAP HANA". Retrieved 2016-03-09.
56. "Hadoop Summit 2014 – Big Data Keeps Getting Bigger". Pentaho. Retrieved 2016-03-09.
57. "Hadoop Summit Europe 2015, 15th-16th April 2015". Lanyrd. Retrieved 2016-03-09.
58. Chouraria, Harsh (21 October 2012). "MR2 and YARN Briefly Explained". *cloudera.com*. Cloudera. Retrieved 23 October 2013.
59. "HDFS User Guide". Hadoop.apache.org. Retrieved 2014-09-04.
60. "Running Hadoop on Ubuntu Linux System(Multi-Node Cluster)".
61. "Running Hadoop on Ubuntu Linux (Single-Node Cluster)". Retrieved 6 June 2013.
62. Evans, Chris (Oct 2013). "Big data storage: Hadoop storage basics". *computerweekly.com*. Computer Weekly. Retrieved 21 June 2016. "HDFS is not a file system in the traditional sense and isn't usually directly mounted for a user to view"
63. deRoos, Dirk. "Managing Files with the Hadoop File System Commands". *dummies.com*. For Dummies. Retrieved 21 June 2016.
64. "HDFS Architecture". Retrieved 1 September 2013.
65. Pessach, Yaniv (2013). "Distributed Storage" (Distributed Storage: Concepts, Algorithms, and Implementations ed.). Amazon.com
66. "Version 2.0 provides for manual failover and they are working on automatic failover:". Hadoop.apache.org. Retrieved 30 July 2013.
67. "Improving MapReduce performance through data placement in heterogeneous Hadoop Clusters" (PDF). Eng.auburn.ed. April 2010.
68. "Mounting HDFS". Retrieved 2016-08-05.
69. Shafer, Jeffrey; Rixner, Scott; Cox, Alan. "The Hadoop Distributed Filesystem: Balancing Portability and Performance" (PDF). Rice University. Retrieved 2016-09-19.
70. Mouzakitis, Evan. "How to Collect Hadoop Performance Metrics". Retrieved 2016-10-24.
71. "HDFS Users Guide – Rack Awareness". Hadoop.apache.org. Retrieved 2013-10-17.
72. "Cloud analytics: Do we really need to reinvent the storage stack?" (PDF). IBM. June 2009.
73. "HADOOP-6330: Integrating IBM General Parallel File System implementation of Hadoop Filesystem interface". IBM. 2009-10-23.

74. "HADOOP-6704: add support for Parascale filesystem". Parascale. 2010-04-14.
75. "HDFS with CloudIQ Storage". Appistry,Inc. 2010-07-06.
76. "High Availability Hadoop". HP. 2010-06-09.
77. job (http://hadoop.apache.org/common/docs/current/commands_manual.html#job) Archived (https://web.archive.org/web/20110817053520/http://hadoop.apache.org/common/docs/current/commands_manual.html#job) August 17, 2011, at the Wayback Machine.
78. "Refactor the scheduler out of the JobTracker". *Hadoop Common*. Apache Software Foundation. Retrieved 9 June 2012.
79. Jones, M. Tim (6 December 2011). "Scheduling in Hadoop". *ibm.com*. IBM. Retrieved 20 November 2013.
80. Hadoop Fair Scheduler Design Document (https://svn.apache.org/repos/asf/hadoop/common/branches/MAPREDUCE-233/src/contrib/fairscheduler/designdoc/fair_scheduler_design_doc.pdf)
81. "CapacityScheduler Guide". Retrieved 31 December 2015.
82. "Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming" (PDF). IEEE. May 2016.
83. " "How 30+ enterprises are using Hadoop", in DBMS2". Dbms2.com. 10 October 2009. Retrieved 2013-10-17.
84. "Yahoo! Launches World's Largest Hadoop Production Application". *Yahoo*. 19 February 2008. Retrieved 31 December 2015.
85. "Hadoop and Distributed Computing at Yahoo!". Yahoo!. 2011-04-20. Retrieved 2013-10-17.
86. "HDFS: Facebook has the world's largest Hadoop cluster!". Hadoopblog.blogspot.com. 2010-05-09. Retrieved 2012-05-23.
87. "Under the Hood: Hadoop Distributed File system reliability with Namenode and Avatarnode". Facebook. Retrieved 2012-09-13.
88. "Under the Hood: Scheduling MapReduce jobs more efficiently with Corona". Facebook. Retrieved 2012-11-09.
89. "Altior's AltraSTAR – Hadoop Storage Accelerator and Optimizer Now Certified on CDH4 (Cloudera's Distribution Including Apache Hadoop Version 4)" (Press release). Eatontown, NJ: Altior Inc. 2012-12-18. Retrieved 2013-10-30.
90. "What is Hadoop?".
91. "Hadoop". Azure.microsoft.com. Retrieved 2014-07-22.
92. "ibm-biginsights-on-cloud".
93. "Oracle's cloud analytics platform comprises several tools". Retrieved 8 April 2016.
94. "HDInsight | Cloud Hadoop". Azure.microsoft.com. Retrieved 2014-07-22.
95. 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.
96. Gottfrid, Derek (1 November 2007). "Self-service, Prorated Super Computing Fun!". *The New York Times*. Retrieved 4 May 2010.
97. "AWS | Amazon Elastic MapReduce (EMR) | Hadoop MapReduce in the Cloud". Aws.amazon.com. Retrieved 2014-07-22.
98. "Amazon Elastic MapReduce Developer Guide" (PDF). Retrieved 2013-10-17.
99. "Amazon EC2 Spot Instances". Aws.amazon.com. Retrieved 2014-07-22.
100. "Amazon Elastic MapReduce Now Supports Spot Instances". Amazon.com. 2011-08-18. Retrieved 2013-10-17.
101. "Amazon Elastic MapReduce FAQs". Amazon.com. Retrieved 2013-10-17.
102. Using Spot Instances with EMR (https://www.youtube.com/watch?v=66rfnFA0jpM) on YouTube
103. "Cloud Computing Services and Managed Services - CenturyLink Cloud".
104. "Managed Cloudera".
105. "Hadoop Simplified: Managed Cloudera &amp; CenturyLink Cloud - CenturyLink Cloud".
106. "Hadoop on Google Cloud Platform".
107. "Google Cloud Dataproc Official Website".
108. "Hadoop on Google Cloud Platform - Command-Line Deployment".
109. "Cloudera now Certified on Google Cloud Platform".
110. "HDP on Google Cloud Platform".
111. "MapR Google Cloud Platform".
112. "Why the Pace of Hadoop Innovation Has to Pick Up". Gigaom.com. 2011-04-25. Retrieved 2013-10-17.
113. "Defining Hadoop". Wiki.apache.org. 2013-03-30. Retrieved 2013-10-17.
114. "Defining Hadoop Compatibility: revisited". Mail-archives.apache.org. 2011-05-10. Retrieved 2013-10-17.
115. "Apache Accumulo User Manual: Security". *apache.org*. Apache Software Foundation. Retrieved 2014-12-03.

Bibliography

- Lam, Chuck (July 28, 2010). *Hadoop in Action* (1st ed.). Manning Publications. p. 325. ISBN 1-935-18219-6.
- Venner, Jason (June 22, 2009). *Pro Hadoop* (1st ed.). Apress. p. 440. ISBN 1-430-21942-4.
- White, Tom (June 16, 2009). *Hadoop: The Definitive Guide* (1st ed.). O'Reilly Media. p. 524. ISBN 0-596-52197-9.

External links

- Official website (http://hadoop.apache.org)
- Apache Hadoop popular APIs in GitHub (http://apiwave.com/java/api/org.apache.hadoop)
- Introducing Apache Hadoop: The Modern Data Operating System (http://www.stanford.edu/class/ee380/Abstracts/111116.html) – a lecture given at Stanford University by Co-Founder and CTO of Cloudera, Amr Awadallah (video archive (http://ee380.stanford.edu/cgi-bin/videologger.php?target=111116-ee380-300.asx)) (YouTube (https://www.youtube.com/watch?v=d2xeNpfzsYI))

- Hadoop with Philip Zeyliger, Software Engineering Radio, IEEE Computer Society, March 8 2010 (http://www.se-radio.net/2010/03/episode-157-hadoop-with-philip-zeyliger/)
- The Key Role Hadoop Plays in Business Intelligence and Data Warehousing (http://online.sju.edu/resource/engineering-technology/key-role-hadoop-plays-in-business-intelligence)

Categories: Apache Software Foundation | Big data products | Cloud infrastructure | Distributed file systems | Free software for cloud computing | Free software programmed in Java (programming language) | Free system software | Hadoop | Software using the Apache license