Hadoop

Source

Alessandro Rezzani, *Big Data - Architettura, tecnologie e metodi per l'utilizzo di grandi basi di dati*, Apogeo Education, ottobre 2013

wikipedia

Hadoop

- Apache Hadoop is an open-source software framework for storage and large scale processing of data-sets on clusters of commodity hardware.

Cluster of machines running Hadoop at Yahoo!
(Source: Yahoo!)

Hadoop

- 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.
  - Hadoop YARN - a resource-management platform
  - Hadoop MapReduce - a programming model for large scale data processing.

Hadoop

Developer(s)	Apache Software Foundation
Stable release	2.2 / October 15, 2013[1]
Preview release	2.1.0-beta / August 25, 2013[1]
Development status	Active
Written in	Java
Operating system	Cross-platform
Type	Distributed File System
License	Apache License 2.0
Website	hadoop.apache.org

Hadoop

- 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.
- Hadoop's MapReduce and HDFS components originally derived respectively from Google's MapReduce and Google File System (GFS) papers.

Hadoop

- 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, and others
- 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.
- Apache Pig, Apache Hive among other related projects expose higher level user interfaces like Pig latin and a SQL variant respectively.

Hadoop

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

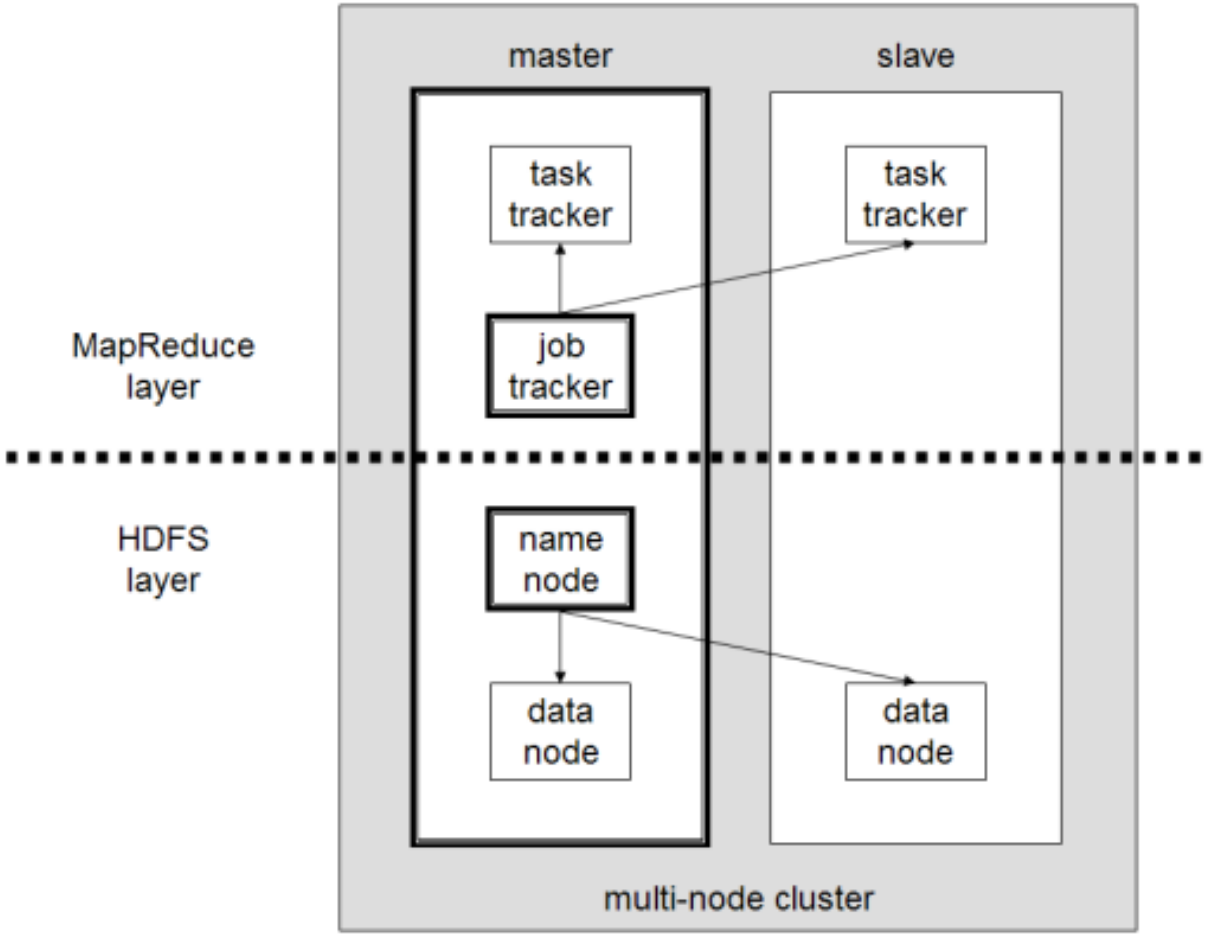
Architecture

- 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.

Architecture

- A small Hadoop cluster includes a single master and multiple worker nodes.
- The master node consists of 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.
- Hadoop requires Java Runtime Environment (JRE) 1.6 or higher.
- The standard start-up and shutdown scripts require Secure Shell (ssh) to be set up between nodes in the cluster.

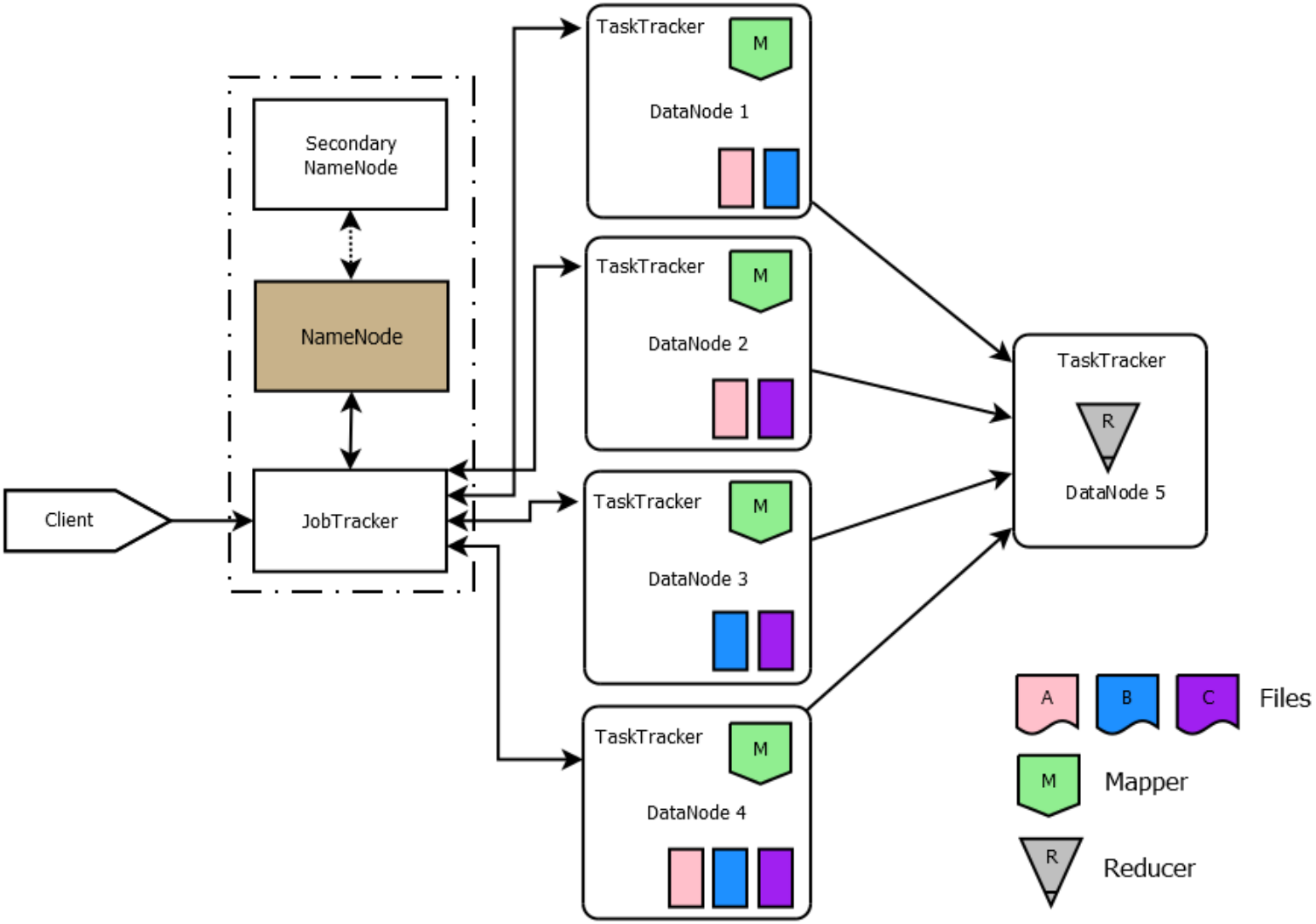
Architecture



Architecture

- 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
- This prevents file-system corruption and reduce 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 is replaced by the file-system-specific equivalent.

Typical Components of a Hadoop Cluster



rohitmenon.com

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.
- Each Hadoop instance typically has a single namenode; a cluster of datanodes form the HDFS cluster.
- Each datanode serves up blocks of data over the network using a block protocol specific to HDFS.

Hadoop distributed file system

- The file system uses the TCP/IP layer for communication. Machines use Remote procedure call (RPC) to communicate between each other
- HDFS stores large files (typically in the range of gigabytes to terabytes) across multiple machines.
- It achieves reliability by replicating the data across multiple hosts, and hence does theoretically not require RAID storage on hosts (but to increase I/O performance some RAID configurations are still useful).

Hadoop distributed file system

- 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

Hadoop distributed file system

- HDFS added the high-availability capabilities, as announced for release 2.0 in May 2012, allowing the main metadata server (the NameNode) to be failed over manually to a backup server in the event of failure.
- The project has also started developing automatic fail-over.

Hadoop distributed file system

- 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

Hadoop distributed file system

- HDFS was designed for mostly immutable files and may not be suitable for systems requiring concurrent write-operations.
- Another limitation of HDFS is that it cannot be mounted directly by an existing operating system.
- Getting data into and out of the HDFS file system, an action that often needs to be performed before and after executing a job, can be inconvenient.
- A Filesystem in Userspace (FUSE) virtual file system has been developed to address this problem, at least for Linux and some other Unix systems.

Hadoop distributed file system

- Files are organized as sequences of blocks of the same dimension (typically 64 MB or 128 MB), redundant on more than one node
- Block dimensions and replica numbers can be configured per file
- Read request are answered by nodes closer to the client

Hadoop distributed file system

- 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 WebHDFS REST API, the command-line interface, or browsed through the HDFS-UI webapp over HTTP.

WebHDFS REST API

http://<HOST>:<PORT>/webhdfs/v1/<PATH>?op=<OPERATION>

- <HOST>:<PORT>: server location
- <PATH>: file or folder location
- <OPERATION>: cat, get, put, ls, mkdir, rm, rmr, copyFromLocal, copyToLocal, count
- Paths in HFS start with hdfs://namenode where namenode is the IP and port local of the NameNode.
- If the prefix is omitted, the system uses the value specified in a configuration file, core-site.xml. If undefined also there, the default is the local file system

Shell HDFS

- Execute commands with

hdfs dfs –option

- Examples
- cat: copies files to standard output

hdfs dfs –cat hdfs://namenode /user/hadoop/file

- get: copies a file from HDFS to the local FS

hdfs dfs –get hdfs://namenode /user/hadoop/file file

- put copies a file from local FS to HDFS

hdfs dfs –put file hdfs://namenode /user/hadoop/file

- ls: returns information on a file or folder

hdfs dfs –ls file hdfs://namenode /data/test

Hadoop command line

hadoop [--config confdir] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]

- Commands
- jar: executes a jar file
- job: interacts with MapReduce jobs
- jobtracker: executes a JobTracker
- tasktracker: executes a TaskTracker
- namenode: executes a NameNode
- datanode: executes a DataNode

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.

JobTracker and TaskTracker

- 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.

JobTracker and TaskTracker

- A heartbeat is sent from the TaskTracker to the JobTracker every few minutes to check its status.
- The JobTracker and TaskTracker status and information can be viewed from a web browser.

Known limitations

- 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 JobTracker 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

Scheduling

- By default Hadoop uses FIFO, and optional 5 scheduling priorities to schedule jobs from a work queue

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
  - 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.

Commercial applications

- 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

- The Yahoo! Search Webmap is a Hadoop application that runs on a more than 10,000 core Linux cluster and produces data that is used in every Yahoo! Web search query
- There are multiple Hadoop clusters at Yahoo! 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.

Prominent users

- On June 10, 2009, Yahoo! made the source code of the version of Hadoop it runs in production available to the public
- In 2010 Facebook claimed that they had the largest Hadoop cluster in the world with 21 PB of storage.
- On June 13, 2012 they announced the data had grown to 100 PB
- On November 8, 2012 they announced the warehouse grows by roughly half a PB per day
- As of 2013, Hadoop adoption is widespread. For example, more than half of the Fortune 50 uses Hadoop