1. **HDFS:**

* The Hadoop Distributed File System (HDFS) is the primary storage system used by [Hadoop](http://searchcloudcomputing.techtarget.com/definition/Hadoop) applications.
* Since HDFS is deployed on low-cost commodity [hardware](http://searchcio-midmarket.techtarget.com/definition/hardware), server failures are common, so the file system is designed to be highly [fault-tolerant](http://searchcio-midmarket.techtarget.com/definition/fault-tolerant) by facilitating the rapid transfer of data between compute nodes and enabling Hadoop systems to continue running if a [node](http://searchnetworking.techtarget.com/definition/node) fails decreasing the risk of [catastrophic failure](http://searchwindowsserver.techtarget.com/definition/catastrophic-failure), even if the numerous nodes fail.
* It detects and compensates for hardware issues, including disk problems and server failure.
* It stores files across the collection of servers in a cluster.
* Files are decomposed into blocks and each block is written to more than one of the servers. As a result, the data on nodes that crash can be found elsewhere within a cluster, which allows processing to continue while the failure is resolved.

*HDFS has been designed keeping in view of the following features:*

* Files that are megabytes, gigabytes, terabytes, or petabytes of size.
* HDFS is built around the idea that data is written once but read many times. A dataset is copied from source and then analysis is done on that dataset over time.
* Hadoop does not require expensive, highly reliable hardware as it is designed to run on clusters of commodity hardware.

One hard drive in 1990 could store 1,370 MB of data and had a

Now, with1-terabyte drive transfer speed is around 100 MB/s.

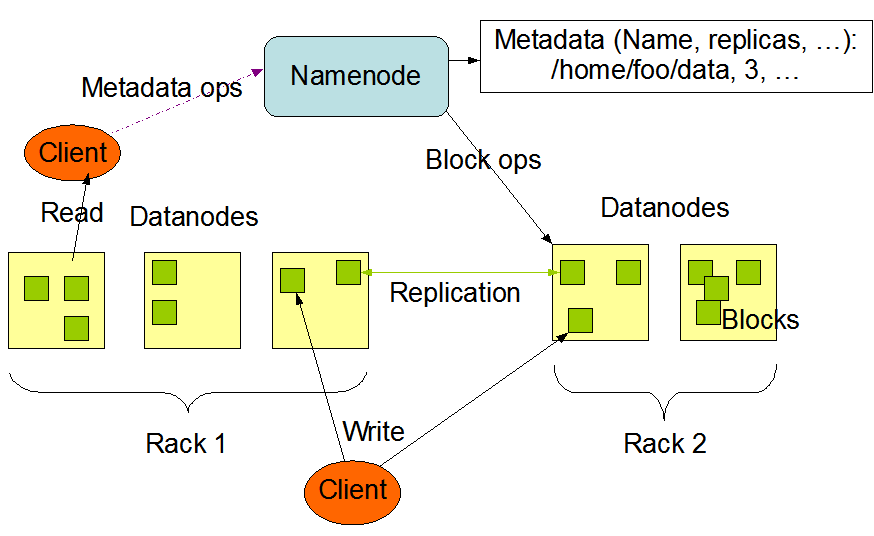
But it takes more than two and a half hours to read all the data off the disk.  
If we had 100 drives, each holding one hundredth of the data, then working in parallel, we could read the data in under two minutes. This is very much similar to distribution of work in any firm.

* It uses a master/slave architecture, with each cluster consisting of a single Name Node that manages file system operations and supporting Data Nodes that manage data storage on individual compute nodes.

***ARCHTICTURE:***

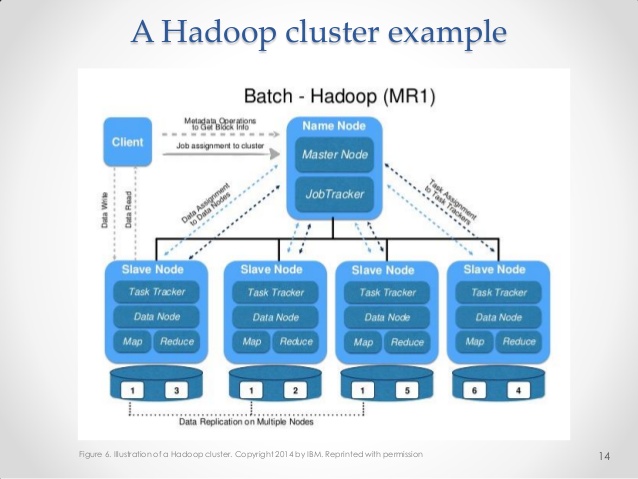
* An HDFS cluster consists of a single NameNode—a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per computer node in the cluster, which manage storage attached to the nodes that they run on.
* The client library manages all communication from the application to the NameNode and the DataNode.
* Each computer has its own file system and information *about* an HDFS file, the metadata which is managed by the NameNode and persistent information is stored in the NameNode’s host file system.
* The information *contained* in an HDFS file is managed by a DataNode and stored on the DataNode’s host computer file system.
* The NameNode executes HDFS file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The list of HDFS files belonging to each block, the current location of the block replicas on the DataNodes, the state of the file, and the access control information is the metadata for the cluster and is managed by the NameNode.
* The DataNodes are responsible for serving read and write requests from the HDFS file system’s clients. The DataNodes also perform block replica creation, deletion, and replication upon instruction from the NameNode. The DataNodes are the arbiter of the state of the replicates and they report this to the NameNode.
* 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 client sends data directly to and reads directly from DataNodes so that client data never flows through the NameNode.

***The figure below shows how blocks are replicated on different DataNodes:***

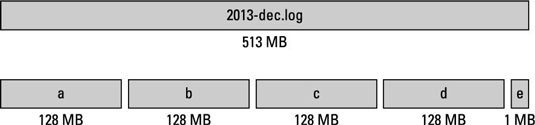


1. **Hadoop Clusters:**

* In a computer system, a cluster is a group of servers and other resources that act like a single system and enable high availability and, in some cases, load balancing and parallel processing.
* A [Hadoop](http://searchcloudcomputing.techtarget.com/definition/Hadoop) cluster is a special type of computational [cluster](http://searchexchange.techtarget.com/definition/cluster) designed specifically for storing and analyzing huge amounts of [unstructured data](http://searchbusinessanalytics.techtarget.com/definition/unstructured-data) in a [distributed computing](http://whatis.techtarget.com/definition/distributed-computing) environment.
* Clusters run Hadoop's [open sourc](http://searchenterpriselinux.techtarget.com/definition/open-source)e distributed processing software on low-cost [commodity computers](http://whatis.techtarget.com/definition/commodity-computer).
* Hadoop clusters are known for boosting the speed of data analysis applications. They also are highly scalable: If a cluster's processing power is overwhelmed by growing volumes of [data](http://searchdatamanagement.techtarget.com/definition/data), additional cluster nodes can be added to increase throughput. Hadoop clusters also are highly resistant to failure because each piece of data is copied onto other cluster nodes, which ensures that the data is not lost if one node fails.
* 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.
* 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.



**HDFS blocks:  
**

* Hadoop distributed file system stores the data in terms of blocks.
* One file can contain many blocks. These blocks in a local file system are nearly 512 bytes and are not necessarily continuous.
* For HDFS, since it is designed for large files, the block size is 128 MB by default. Moreover, it gets blocks of local file system contiguously to minimize the head seek time.
* The blocks are of fixed size, so it is very easy to calculate the number of blocks that can be stored on a disk.
* HDFS block concept simplifies the storage of the datanodes. The datanodes doesn’t need to concern about the blocks metadata data like file permissions etc. The namenode maintains the metadata of all the blocks.
* As the file is chunked into blocks, it is easy to store a file that is larger than the disk size as the data blocks are distributed and stored on multiple nodes in a hadoop cluster.
* Hadoop framework replicates each block across multiple nodes (default replication factor is 3). In case of any node failure or block corruption, the same block can be read from another node.
* If the size of the file is less than the HDFS block size, then the file does not occupy the complete block storage.  
  ***example*** if a file size of 513 MB, data is stored as shown below:   
  

Blocks of file in local FS:  


Blocks of file in HDFS:  


*Few HDFS commands:*   
  
# Print the Hadoop version  
hadoop version  
  
# List the contents of the root directory in HDFS  
hadoop fs -ls /  
  
# To view the contents of your text file test.txt which is present in your hadoop directory.  
hadoop fs -cat hadoop/test.txt  
  
#  cp is used to copy files between directories present in HDFS  
hadoop fs -cp /user/acadgild/\*.txt /user/acadgild/Hadoop  
  
# List all the hadoop file system shell commands  
hadoop fs  
  
# Ask for help!  
hadoop fs -help