**HDFS:**

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is now an Apache Hadoop subproject.

When HDFS takes in data, it breaks the information down into separate pieces and distributes them to different nodes in a cluster, allowing for [parallel processing](http://whatis.techtarget.com/definition/parallel-processing-software). The file system also copies each piece of data multiple times and distributes the copies to individual nodes, placing at least one copy on a different [server rack](http://whatis.techtarget.com/definition/rack-server-rack-mounted-server) than the others. 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 is built to support applications with large data sets, including individual files that reach into the terabytes. It uses a master/slave architecture, with each cluster consisting of a single NameNode that manages file system operations and supporting DataNodes that manage data storage on individual compute nodes.

**Hadoop Cluster:**

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.

Such 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). Typically one machine in the cluster is designated as the NameNode and another machine the as JobTracker; these are the masters. The rest of the machines in the cluster act as both DataNode and TaskTracker; these are the slaves. Hadoop clusters are often referred to as "shared nothing" systems because the only thing that is shared between nodes is the network that connects them.

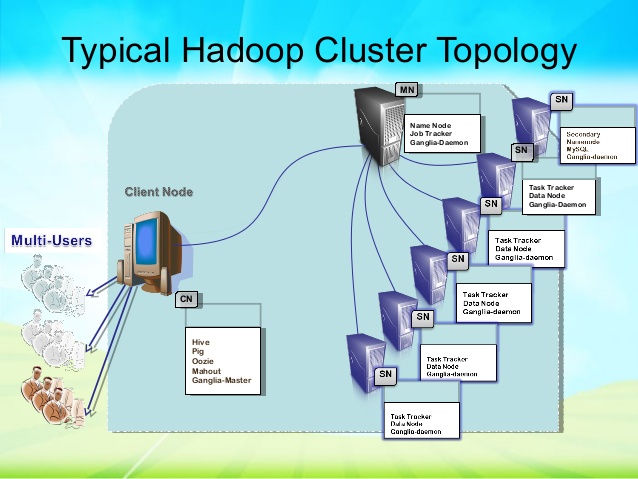
Hadoop clusters are known for boosting the speed of data analysis applications. They also are highly scalable: If a cluster's processing power is overwhelmed by growing volumes of [data](http://searchdatamanagement.techtarget.com/definition/data), additional cluster nodes can be added to increase throughput. 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.

As of early 2013, [Facebook](http://whatis.techtarget.com/definition/Facebook) was recognized as having the largest Hadoop cluster in the world. Other prominent users include [Google](http://searchcio-midmarket.techtarget.com/definition/Google), [Yahoo](http://whatis.techtarget.com/definition/Yahoo) and [IBM](http://searchitchannel.techtarget.com/definition/IBM-International-Business-Machines).

A Hadoop cluster is commonly referred to as "shared nothing." One of the things that distinguishes HDFS from some of the more common file systems like NFS and CIFS is its ability to support a distributed computing, shared nothing [architecture](http://www.itreek.spb.ru/docs/docs/hpts85-nothing.pdf).

Shared nothing means almost exactly what it says. In a [distributed computing cluster](http://searchenterpriselinux.techtarget.com/answer/Distributed-computing-clusters-and-high-availability-solutions) composed of parallelized nodes, the only thing that's actually shared is the cluster network that interconnects the compute nodes. Nothing else is shared, including storage, which is implemented as disk-based direct-attached storage ([DAS](http://searchstorage.techtarget.com/definition/direct-attached-storage)). Usually DAS here consists of one set of eight to 10 disks per node configured as RAID or JBOD for maximum performance. Solid-state drives (SSDs) aren't typically used because of cost.

One of the objectives of the shared nothing paradigm is to reduce processing latency. Keep in mind that we want to process queries that grind through an enormous amount of data, often in five seconds or less. So minimizing cluster-wide latency is a critical priority for Hadoop developers and system architects.

****

**HDFS Blocks:**

Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block.

A disk has a block size, which is the minimum amount of data that it can read or write. Filesystems for a single disk build on this by dealing with data in blocks, which are an integral multiple of the disk size block.

Filesystem blocks are typically a few kilobytes in size, whereas disk blocks are normally 512 bytes. This is generally transparent to the filesystem user who is simply reading or writing a file of whatever length. However, there are tools are to perform filesystem maintenance, such as ***df*** and ***fsck,*** that operateon the filesystem block level.

HDFS, too, has a concept of a block, but it is much larger unit—128 MB by default. Like in a filesystem for a single disk, files in HDFS are broken into block-sized chunks, which are stored as independent units.

Unlike a filesystem for a single disk, a file in HDFS that is smaller than a single block does not occupy a full block’s worth of underlying storage.

(For example, a 1 MB file stored with a block size of 128 MB uses 1MB of disk space, not 128 MB).

HDFS blocks are large compared to disk blocks, and the reason is to minimize the cost of seeks. If the block is large enough, the time it takes to transfer the data from the disk can be significantly longer than the time seek to the start of the block. Thus, transferring a large file made up of multiple blocks operates at the disk transfer rate.

Having a block abstraction for a distributed filesystem brings several benefits. The first benefit is the most obvious: a file can be larger than any single disk in the network. There’s nothing that requires the blocks from a file to be stored in on the same disk, so they can take an advantage of any of the disk in the cluster. In fact, it would be possible, if unusual, to store a single file on an HDFS cluster whose blocks filled all the disks in the cluster.

Second, making the unit of abstraction a block rather than a file simplifies the storage subsystem. Simplicity is something to strive for in all systems, but it is especially important for a distributed system in which the failure nodes are so varied. The storage subsystem deals with blocks, simplifying storage management (because blocks are fixed size, it is easy to calculate how many can be stored on given disk) and eliminating metadata concerns (because blocks are just chunks of data to be stored, file metadata such as permissions information does not need to be stored with the blocks, so another system can handle metadata separately).

Furthermore, blocks fit well with the replication for providing fault tolerance and availability. To insure against corrupted blocks and disk and machine failure, each block is replicated to small number of physically separate machines. If a block becomes unavailable, a copy can be read from another location in a way that it is transparent to the client. A block that is no longer available due to corruption or machine failure can be replicated from its alternative locations to other live machines to bring the replication factor back to the normal level.

