Hadoop 2.X components are

1. HDFS
2. YARN
3. **HDFS**

HDFS has many similarities with other distributed file systems, but is different in several respects. One noticeable difference is HDFS's write-once-read-many model , simplifies data coherency, and enables high-throughput access.

Another unique attribute of HDFS is the viewpoint that it is usually better to locate processing logic near the data rather than moving the data to the application space.HDFS rigorously restricts data writing to one writer at a time. Bytes are always appended to the end of a stream, and byte streams are guaranteed to be stored in the order written.

**Benefits of HDFS**

a. Fault tolerance by detecting faults and applying quick, automatic recovery

b. Data access via MapReduce streaming

c. Simple model

d. Efficiency by distributing data and logic to process it in parallel on nodes where data is located.

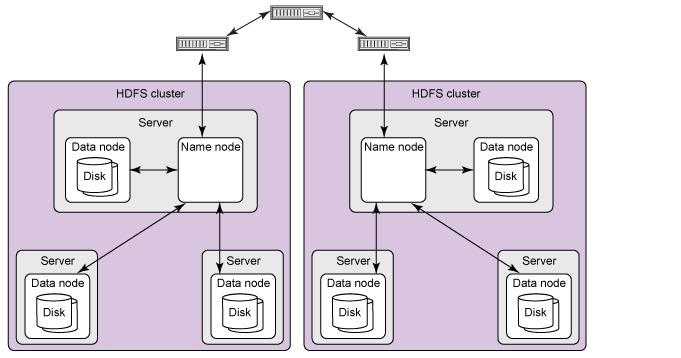
e. Reliability by automatically maintaining multiple copies of data and automatically redeploying processing logic in the event of failures

**HDFS Architecture**

HDFS is comprised of interconnected clusters of nodes where files and directories reside. An HDFS cluster consists of a single node, known as a NameNode, that manages the file system namespace and regulates client access to files. In addition, data nodes (DataNodes) store data as blocks within files.

Within HDFS, a given name node manages file system namespace operations like opening, closing, and renaming files and directories. A name node also maps data blocks to data nodes, which handle read and write requests from HDFS clients. Data nodes also create, delete, and replicate data blocks.

Fig : HDFS Architecture



Name nodes and data nodes are software components designed to run in a decoupled manner on commodity machines across heterogeneous operating systems. HDFS is built using the Java programming language; therefore, any machine that supports the Java programming language can run HDFS. A typical installation cluster has a dedicated machine that runs a name node and possibly one data node. Each of the other machines in the cluster runs one data node.

Data nodes continuously loop, asking the name node for instructions. A name node can't connect directly to a data node; it simply returns values from functions invoked by a data node. Each data node maintains an open server socket so that client code or other data nodes can read or write data. The host or port for this server socket is known by the name node, which provides the information to interested clients or other data nodes.