

HDFS Notes



[More info](#)

G. Brancati



HDFS è un file system distribuito che fornisce uno storage per file enormi con accesso streaming eseguito su commodity hardware.

- Extremely large files: dati nell'ordine di petabyte

- Streaming Data Access Pattern: principio di write-once and read-many-times

- Commodity hardware: hardware non costoso e disponibile nel mercato

HDFS ha un architettura MASTER/SLAVE. Un cluster HDFS consiste in un singolo **NameNode** (chiamato **MasterNode**) ed un numero variabile di **DataNodes** (detti **Slave Node**).

- **MasterNode**: - gestisce gli store e assegna task - Esegue operazioni di filesystem quali creazione, lettura, scrittura - Eseguito su hardware "sicuro".

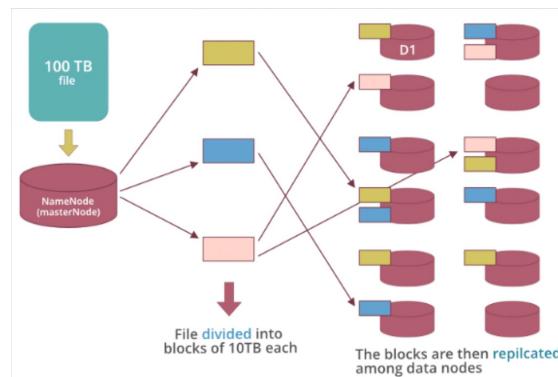
- **SlaveNode**: - Esegue task assegnati dal master - Operazioni di FS indicate dal master - Eseguito su commodity hardware.

Su queste machine cominciano dei learning.

- **Namenodes**:
 - Run on the master node.
 - Store metadata (data about data) like file path, the number of blocks, block IDs, etc.
 - Require high amount of RAM.
 - Store meta-data in RAM for fast retrieval i.e. to reduce seek time. Though a persistent copy of it is kept on disk.
- **DataNodes**:
 - Run on slave nodes.
 - Require high memory as data is actually stored here.

DATA STORAGE

Allora i dati sono distribuiti tra i vari nodi.



ESEMPIO.

Assumiamo che vengono inseriti 100TB. Questi vengono divisi dal MasterNode in **BLOCCI** di 10TB, i quali vengono salvati tra i diversi nodi. Questi blocchi vengono **Duplicati** tra gli slave. Ogni nodo invia info sui blocchi che contiene al master.

Il master node dunque ha info su tutti, la località di ogni nodo + i blocchi che contiene, tutto è gestito da esso.

BLOCCI: I dati vengono divisi in blocchi poiché salvare un solo file risulterebbe difficile, in più permette una lettura e scrittura più semplice e veloce. In più questo permette la **REPLICATION**. Ogni blocco viene salvato in più nodi in modo tale che se un nodo crasha è comunque possibile recuperare i dati (**FAULT TOLERANCE**). Diversamente lo stesso blocco non può essere salvato sullo stesso nodo.

HEARTBEAT: Il segnale che ogni datanode invia al namenode. Se non arriva il datanode è considerato morto.

BALANCING: Se un nodo muore i blocchi su di esso vengono ridistribuiti per tornare al replication set.

REPLICATION: Duplicazione dei nodi tra i vari nodi.

FEATURES

- Storage distribuito - Blocco per velocizzare l'I/O - dati altamente disponibili - Fault Tolerance - altamente affidabile

LIMITI

- Accesso ai dati a bassa latenza. Non si adatta a cosi l'uso low-latency. - Problema piccoli file: inefficiente con file piccoli

FS Shell

HDFS allows user data to be organized in the form of files and directories. It provides a commandline interface called FS shell that lets a user interact with the data in HDFS. The syntax of this command set is similar to other shells (e.g. bash, csh) that users are already familiar with. Here are some sample action/command pairs:

Action	Command
Create a directory named /foodir	bin/hadoop dfs -mkdir /foodir
Remove a directory named /foodir	bin/hadoop dfs -rmr /foodir
View the contents of a file named /foodir/myfile.txt	bin/hadoop dfs -cat /foodir/myfile.txt

FS shell is targeted for applications that need a scripting language to interact with the stored data.

DFSAdmin

The DFSAdmin command set is used for administering an HDFS cluster. These are commands that are used only by an HDFS administrator. Here are some sample action/command pairs:

Action	Command
Put the cluster in SafeMode	bin/hadoop dfsadmin -safeMode enter
Generate a list of DataNodes	bin/hadoop dfsadmin -report
RecCommission or decommission DataNode(s)	bin/hadoop dfsadmin -refreshNodes