Data storage

- ① HDFS
- ② HBase

- Data processing & Analytics
  - Batch
    - map reduce
    - Spark
    - Hive

- Online streaming process
  - storm
  - Spark

- Data Sharding
  - Data partitioned into groups & stored on different machines
    ex. HDFS
    NO SQL DBDS
  - Concurrent processing of data on different machines
  - Speed up processing time.

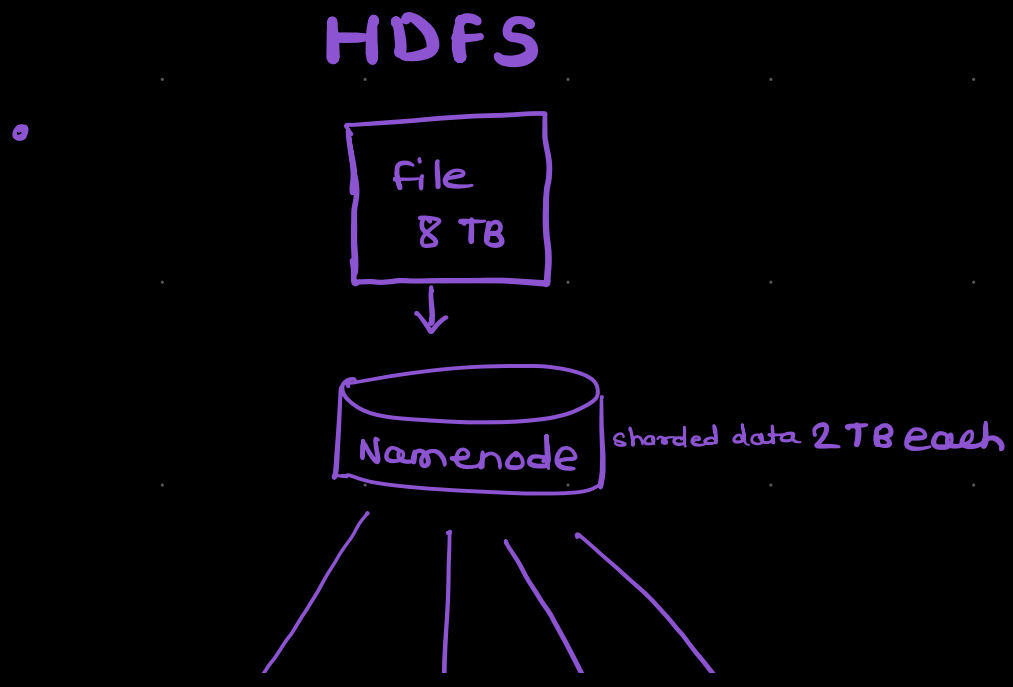
- Distributed Algorithms

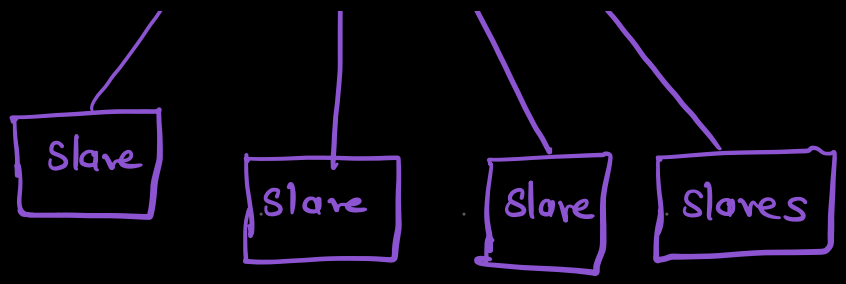
  Set of collaborative tasks which might run concurrently to process data on different machines.

Example -

- Hadoop Map reduce
- Distributed ML

framework to execute
- Hadoop YARN





HDFS uses data sharding to create blocks of data & then it is distributed

Data co-ordinator = Namenode

shard manager = Datanode.

- Namenode Responsibility
  1. – metadata
     - tree structure
     - directory metadata
     - files metadata
  2. location of each block

Stores in this way

- Namespace img files

  fsimage - Latest checkpoint

Datanode Responsibility

- preserve the assigned blocks on its local disk
- Each block is identified using blockpool id & block id
- Management of assigned blocks.

HDFS Availability

- large cluster = High chances of failure

  failure = datanode crash, datablock on that datanode is not available

- If namenode crash HDFS unavailable for client

Two types of redundancy

1. Block
2. Namenode

① Block -

- Replicas are persisted on different m/c
- Number of block replicas are configurable with HDFS-site.xml
- client can read from any copy
- Client write to all copies

② Namenode Redundancy

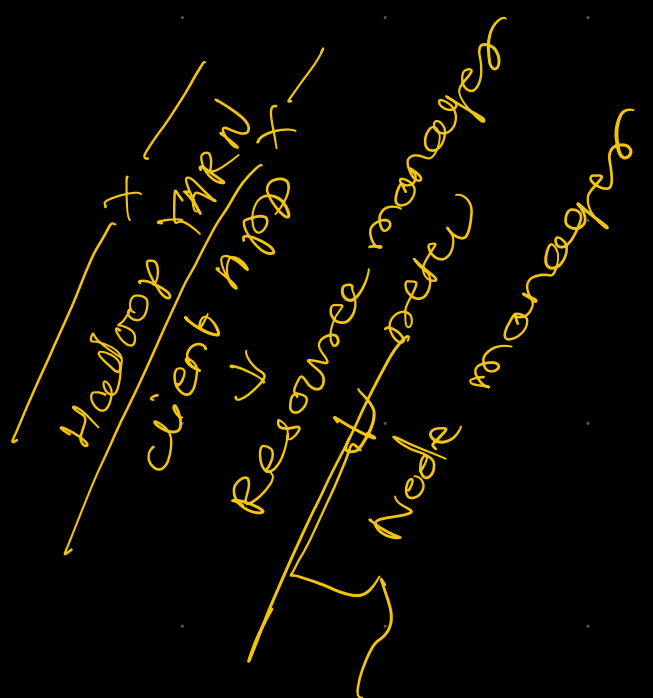
① Additional secondary namenode can be implemented

primary crashed secondary takes over

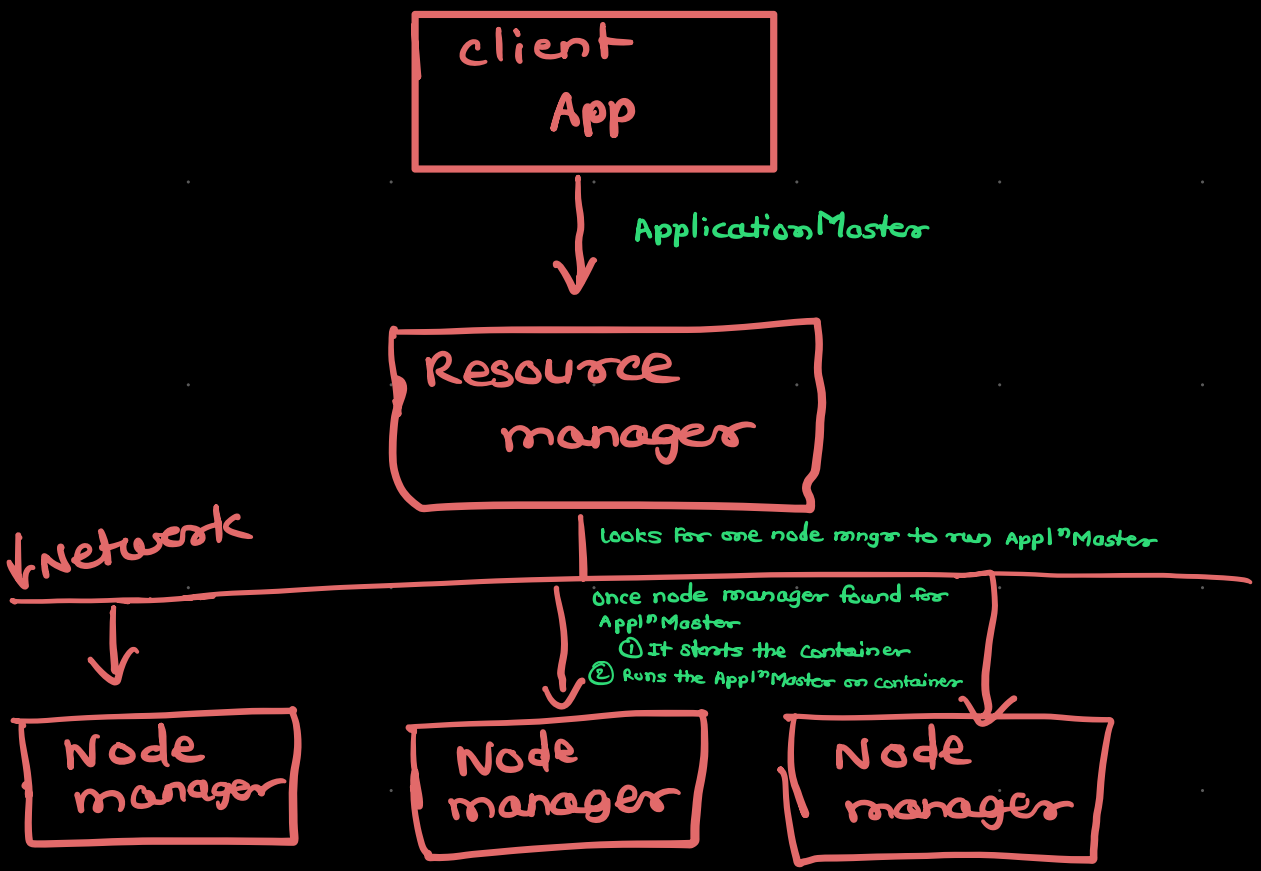
Two types of namenode fail-over

① Pre-release 2 namenode fail

② Release 2 namenode failovers.



Hadoop YARN

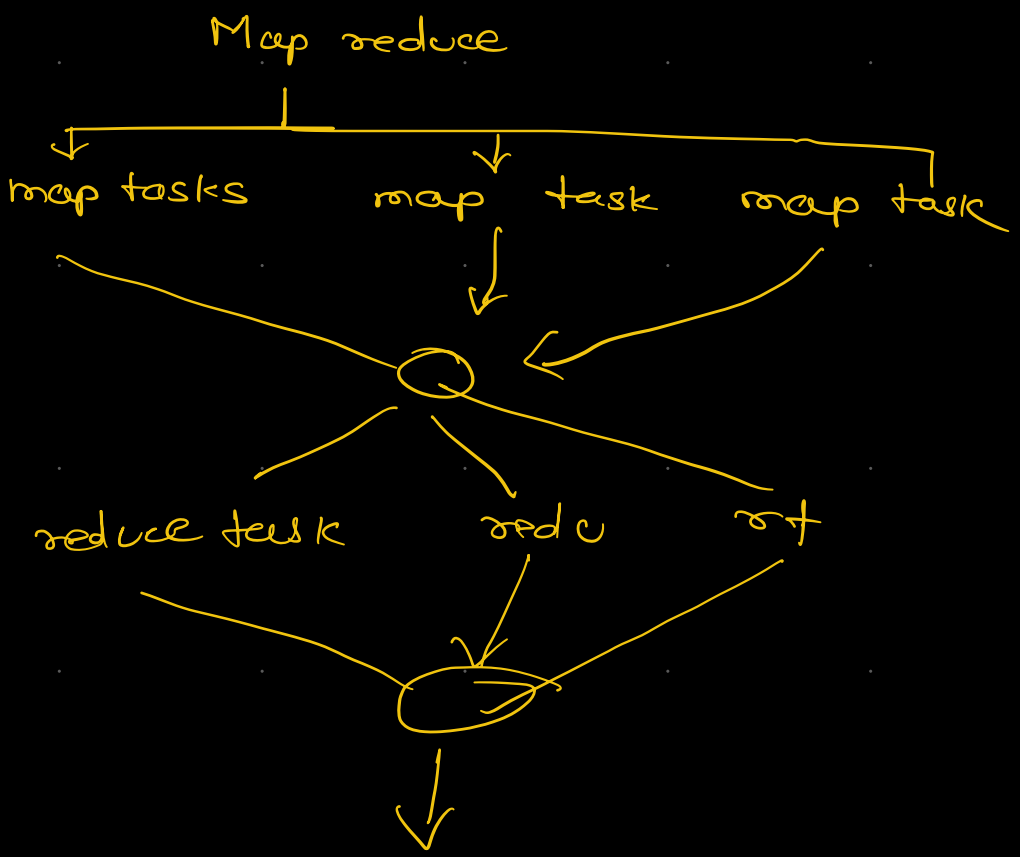


- As YARN uses distributed algorithm It is implemented as YARN Appl^n Master.

Client job sends ApplicationMaster to ⟶
Resource manager
↓
look for node manager to run Appl^n manager
once node manager selected, Resource manager will ask the node manager to

① Start container
② Run ApplicationMaster on container

* YARN Job Scheduling
  ① FIFO scheduler
  ② Capacity scheduler
  ③ fair scheduler



- perform aggregate operations on each group

- Join two files with a given join key.

Two phases

① Map phase

— Retrieve data from local

redistribute the local data

② Reduce phase

Apply analytical logic to the data coming from other