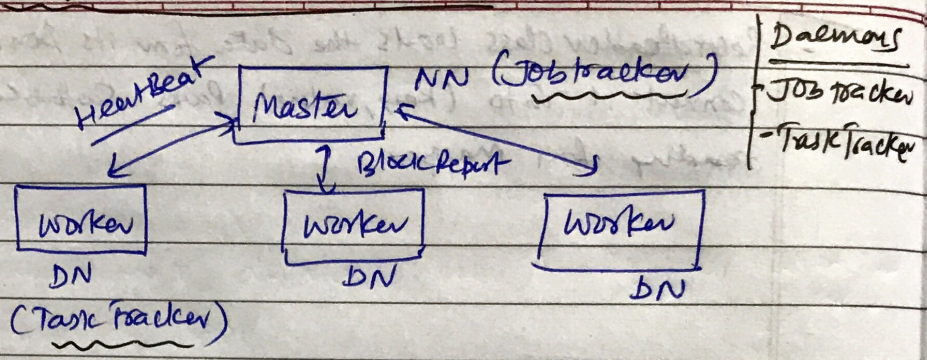
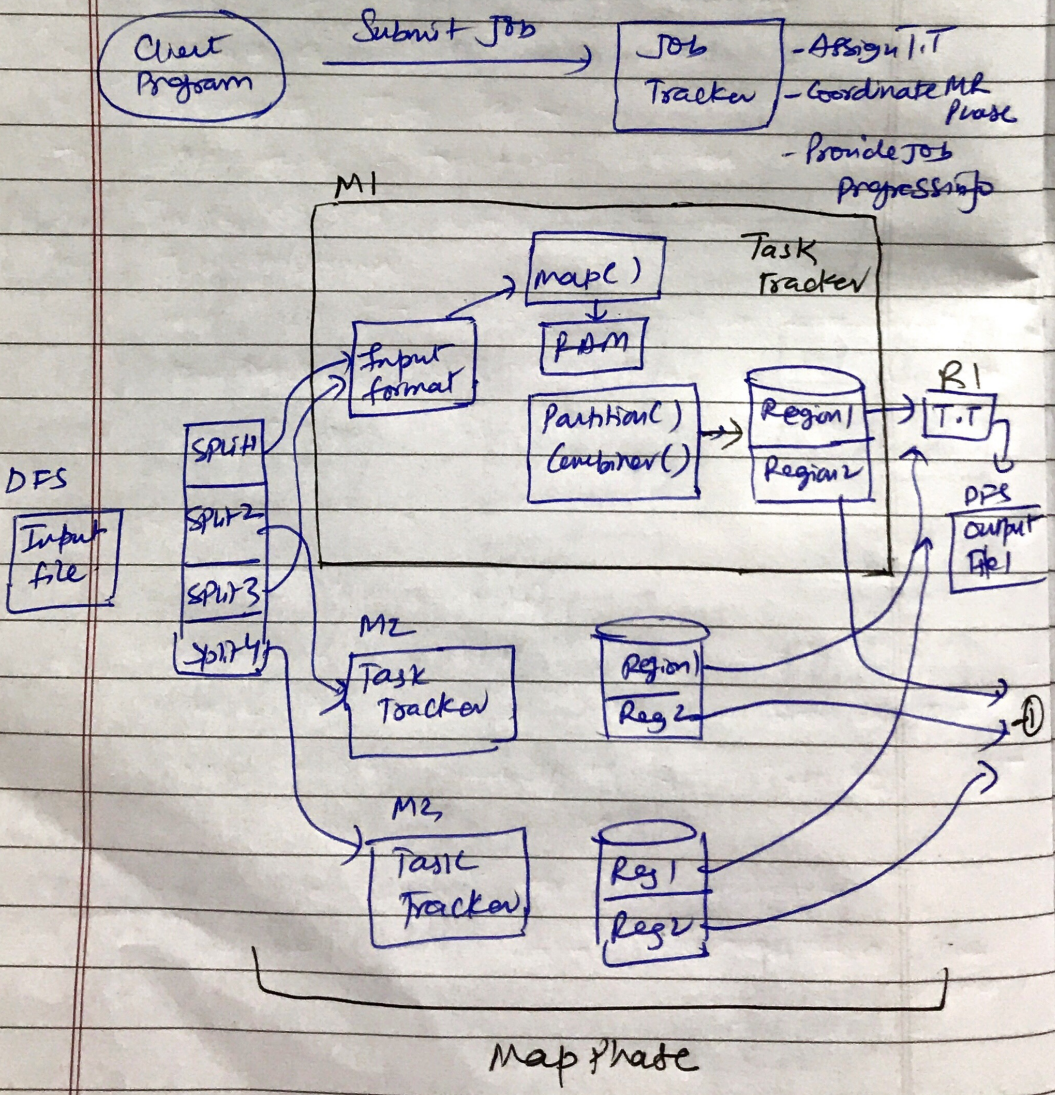
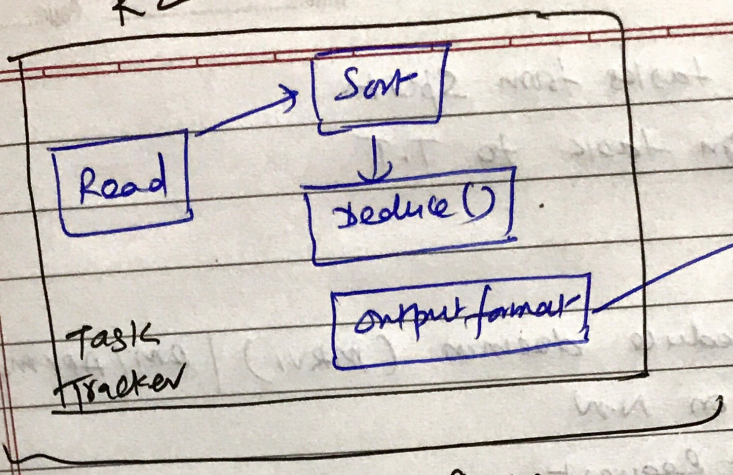
Date: Page:

Map Reduce Architecture

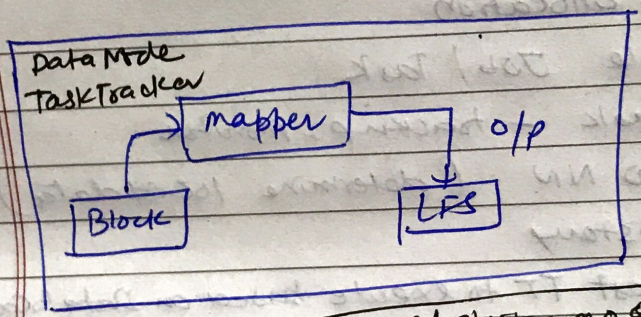




R2 Date: Page:



Reduce Phase



Worker M/C

75%	25%
HDFS	LFS

Data locality

no of split = no of mapper

Workflow

1) Copy input files to DFS
2) Submit Job to client
3) Client get info about input files from NN.
4) Client create Split's of all file (Job)
5) Store metadata about this Job to DFS (job.jar)
6) Client Submit Job to JobTracker

JT 7) JT initialize Job with Job queue.

8) JT read Job files from DFS
9) JT create map & Reduces for Job and input split to mapper.

TT 10) Receive Heart beat from TT to JT. (3sec)
if not receiving wait for more 30sec - declare dead - and update metadata

Date: Page:

11) Pick tasks from split
12) Assign task to T.T

JobTracker

1) MapReduce daemon (MRV1) | RM / AppMaster in MRV2
2) Runs on N.N
3) Client Request
4) Split Job/work - Mapper / Reducer
5) Resource allocation
6) Schedule Job / Task
7) Job / task - tracking - monitor
8) Talk to NN (determine loc of data)
9) Job history
10) Find best TT to execute based on Data locality
11) JobTracker process is Critical for MR execution
12) When JT down, HDFS is still function / MR will be halted

TaskTracker

1) Runs on Data node
2) MapReduce daemon MRV1 | Node Mgr in MRV2
3) MR tasks executed on D.N administer by TT
4) Assign Mapper / Reducer tasks to execute by JT
5) Progress of task to JT.
6) Send heart beat to J.T (Alive)
7) T.T failure is not fatal. (JT Assign to another node)