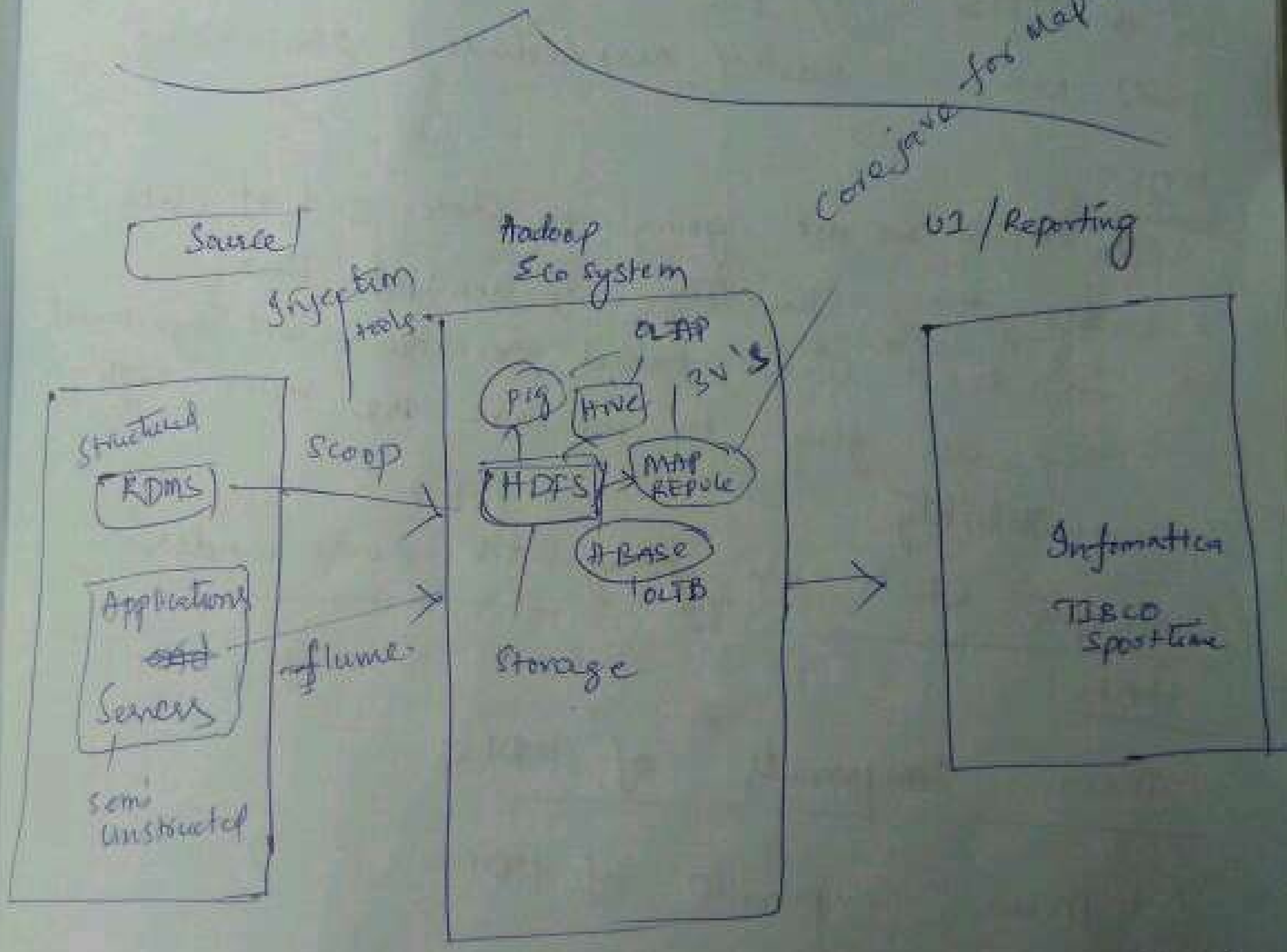
OLAP - Batch processing

1

Oozie

putty
winscp
one note
notepad ++



Hive - Structured (SQL)

Mapreduce - 3V's

pig - (~~NoSQ~~)

OLAP - No Bulk

OLTP - online tractions
live data

Online Analytical P
Processing

Hadoop has two main components. (Storage)

1) HDFS - (Hadoop distributed file systems)

2) MR - Hadoop MapReduce) - processing.

HDFS

In HDFS we are going to store 3v's of data ie unstructure, structure & semi structure.

-> In Read time based on the client Requirement they are going to increase the HDFS high availability

Concepts which are going to cover under HDFS:

Main components of HDFS:

1. Main components of HDFS

* Master slave architecture
* Replicatation factor concept.

3> Rack awarness concept

4> How to read data from HDFS

5> How to write data into HDFS

6> Commands under HDFS.

7> HDFS high availability.

8> HDFS meta data high availability.

9> Relation b/w HDFS & M/R

10> what are the main configuration files of Hadoop.

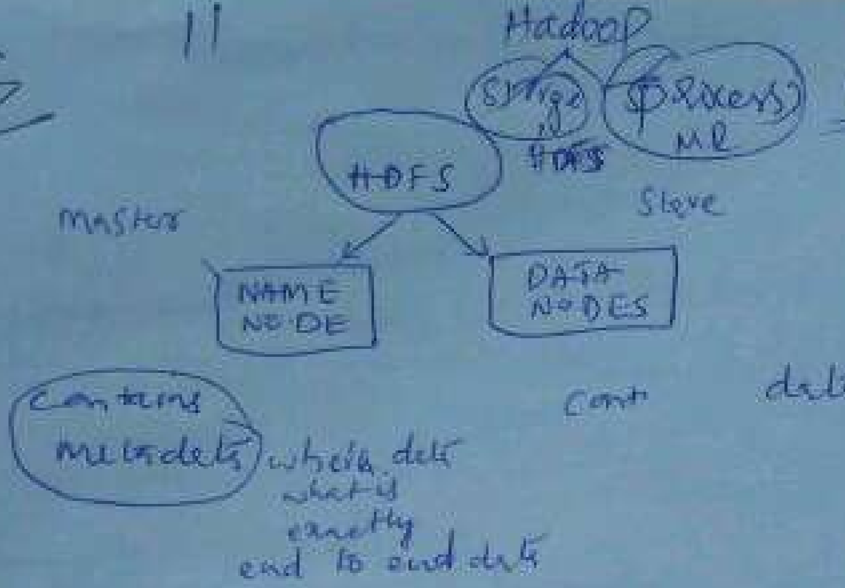
→ **Kerbaroe** is used give security to entire cluster in hortonworks }.

Apache sentry → cloudera } Vendors

4 11

HDFS

13-02-2016



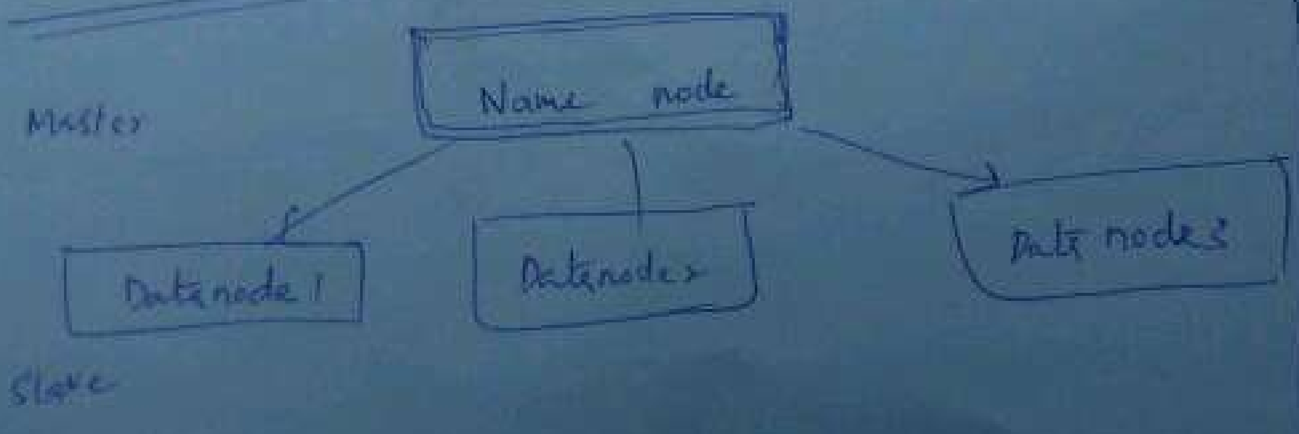
HDFS contains two main ~~nodes~~ nodes i> Name node
ii> data node

i> Name node:
Name node is like master of ~~our~~ hadoop architecture.
It contains meta data ~~also~~ information about our cluster.

ii> Data node:- Data node containg data of our project.

Ex:- my text book.

Architecture of HDFC :-



master-slave architecture

→ Data node depends on fault tolerance to maintain the high availablity ~~of~~ data (scaleout) -

replication factor

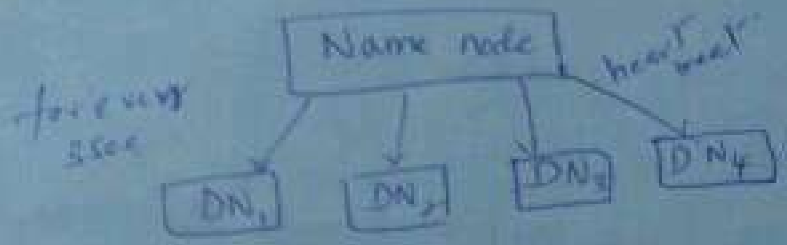
ex: 10 — 3 — (3 times) data nodes.

Ex: 20TB — 500 GB (data node size).
40x3 = 120 (no of nodes).

OLTP
OLAP - MR
hive
pig

- HDFC : Storage
- mapreduce : processing

* Name node and Data node communication :



- for every 3sec it will sends Report all (live or dead)
- for every 10th heart beat contains Block (Report).

10x30 sec. (block report for every 30sec).

Block report
1) How much space is occupied.
2) How much space is left for usage.

Exercise:
Day 1:

1) How we are achiving fault tolerance in hadop?
2) ~~E~~Explain four ~~4~~ V's of hadoop
3) Explain master slave architecture with communication chan[illegible]

14-02-2019

+ Blocks: physical division of data

Blocks in HDFS:

Default block size = 64MB · hadoop 1.x version
default block size = 128MB hadoop 2.x version.

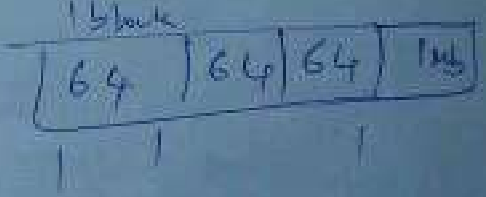
=> Block is a physical division of data as per hadoop 1.x version default block size is 64MB. w.r.t to 2.x version default block size is 128MB.

for Eg:

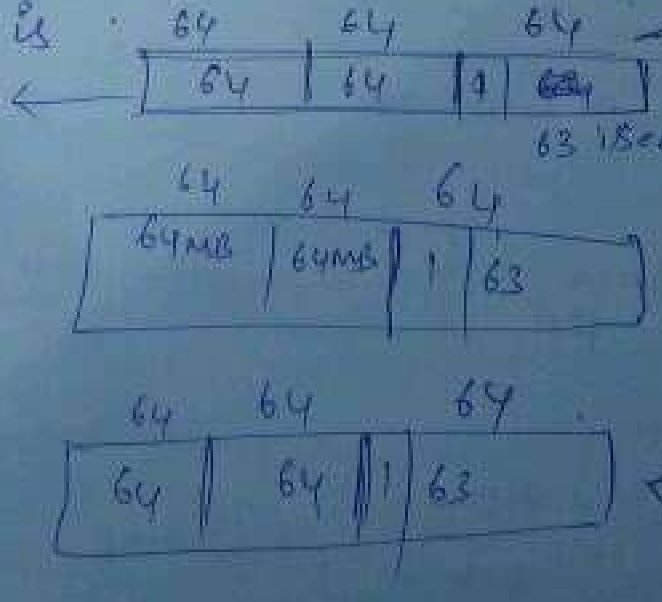
for 250 MB file total size will be

$\frac{250}{64} \times 3 = \frac{750}{64}$ MB

$\frac{250}{64} = \cancel{4} = 12$ blocks.



Suppose file size is 129 MB. Block Representation of this file is .



→ 1st copy.

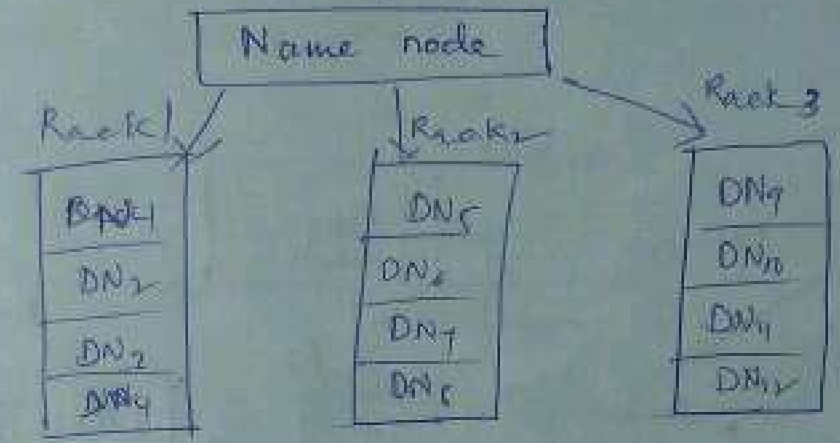
we cannot use this free space of 63 mb for any other storage

← 2nd copy

← 3rd copy

$\frac{63 \times 3}{189}$ MB is wasting.

Rack awarness policy:



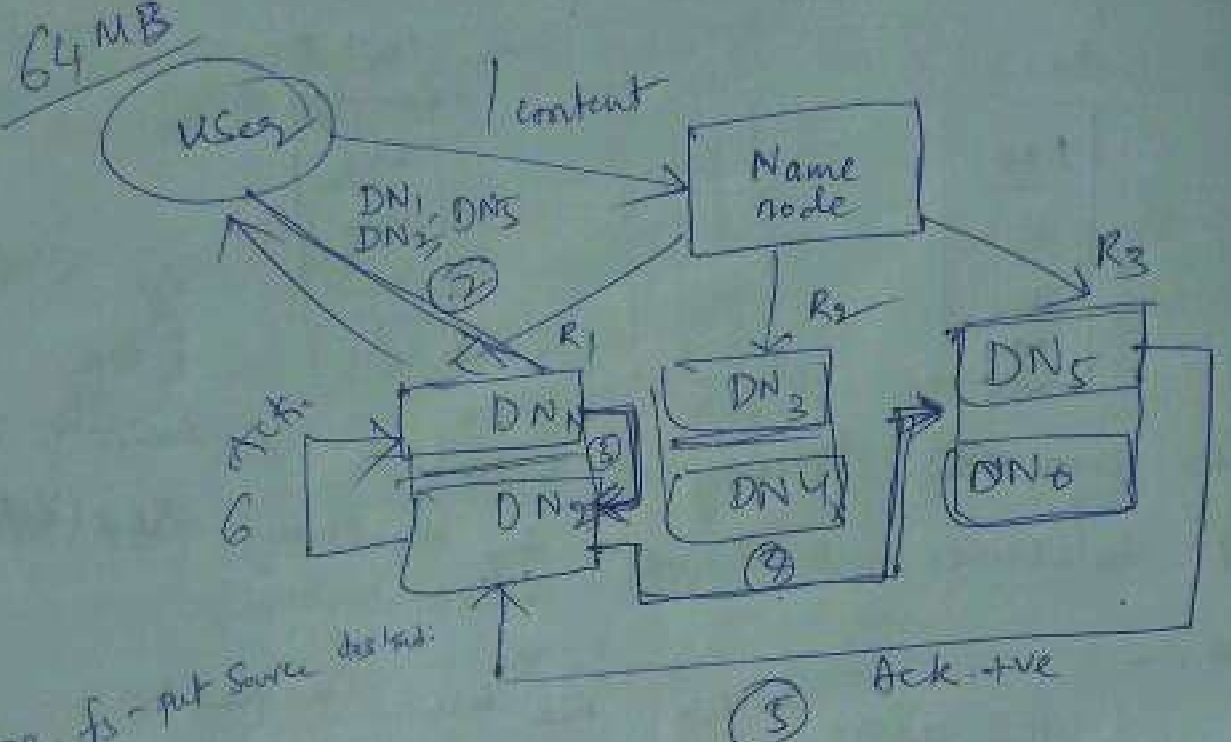
Rack awaness polocly is used to achive fault tolerance: that i.e by maintaing replication factor 3.

In the above fig we have 3 Racks. each rack contains 4 data nodes totally, 4x3 = 12 DN. datanodes

for Every 3 sec every data node will send heart beats to name node.

When we are storing ~~any~~ 64 MB file in to hadoop it has to copy 3 blocks. because it has to satisfy fault tolerance. This 3 blocks should be stored in 2 racks. i.e "2 copies of data should be in 2 data nodes under one rack, another copy should be in another rack."

How to write data into HDFS:



Hadoop fs -put source destination

If we want to store any data in hadoop environment it has to Satisfy 2 features.

(1) Falut tolerance

(2) Rack awarness policy.

⇒ first we have to Contact with name node.

⇒ Name node will give you the data nodes information for storage.

⇒ Based On the metadata which we got from the name node we are storing the data into HDFS.

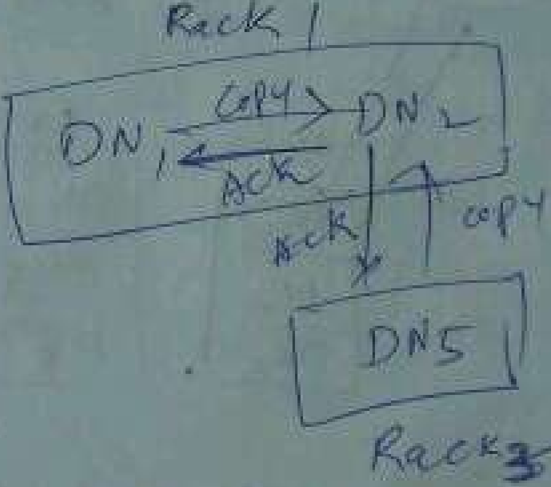
⇒ Above diagram explaing

↳ First we have to Contacte with name node

↳ name node has provided DN, DN2, and DN5 are the nodes which i have to copy.

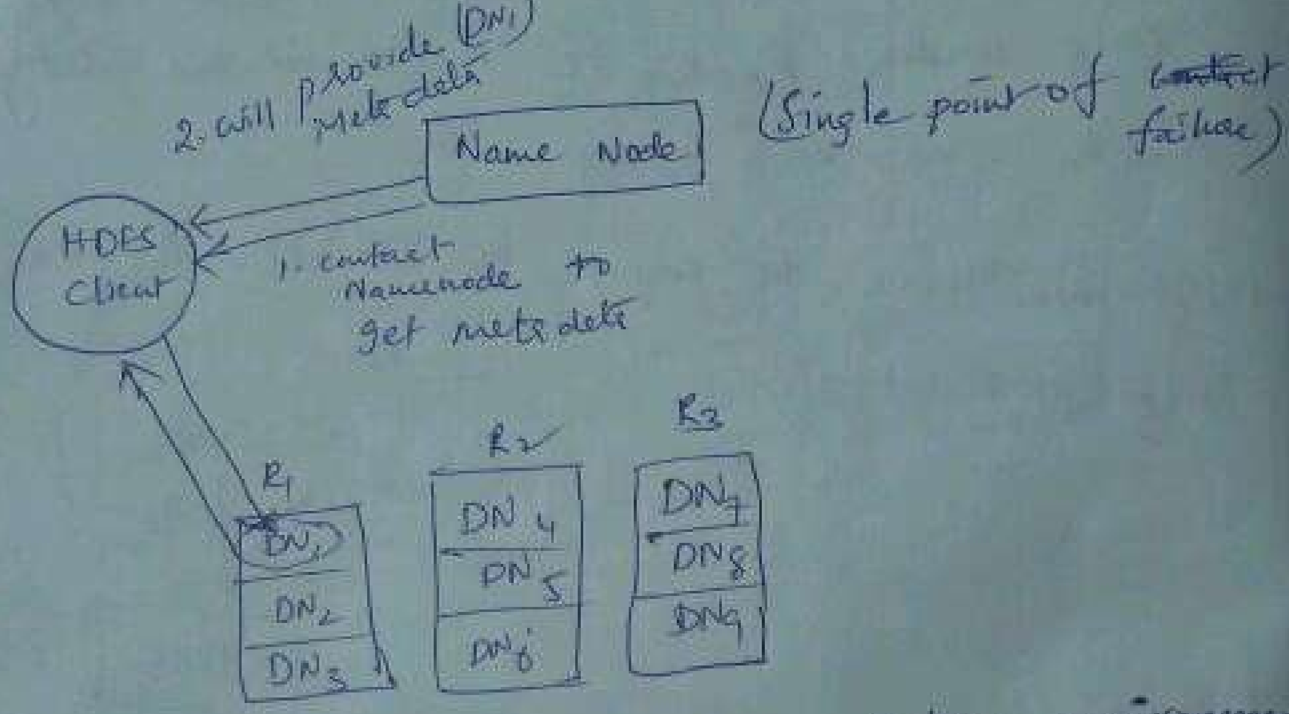
↳ Data Node, DN2 are under 1 Rack i.e R1 DN5 under Rack3 i.e means it has satisfy Rack avarness polecy.

first we have to copy in DN1, DN2 in Rack 1 DN5 in Rack 3.



15.02.17

1) How to Read the data from HDFS :


2. will provide (DN1) meta data
Name Node
(Single point of ~~contact~~ failure)
HDFS Client
1. contact Namenode to get meta data
R1
R2
R3
DN1
DN2
DN3
DN4
DN5
DN6
DN7
DN8
DN9

=> If we want to get data from HDFS environment first point of contact is Name Node. (Master)

=> Then name node w'l give you metadata information about data.

then we can check data in particular data node.

Cluster : collection of data node and Namenode

project size (10TB) = 30TB x 1 TB = 30 DN

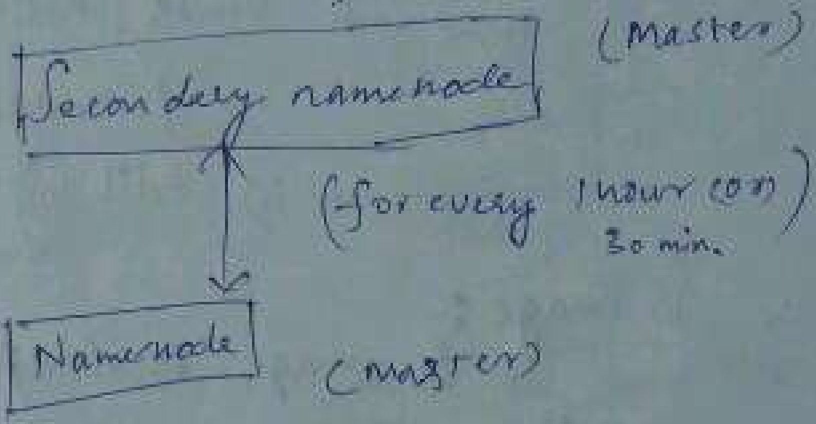
= 30 + 2 (1 NN, 1 SNN)

= 32

project size

10TB → DN1 → TB

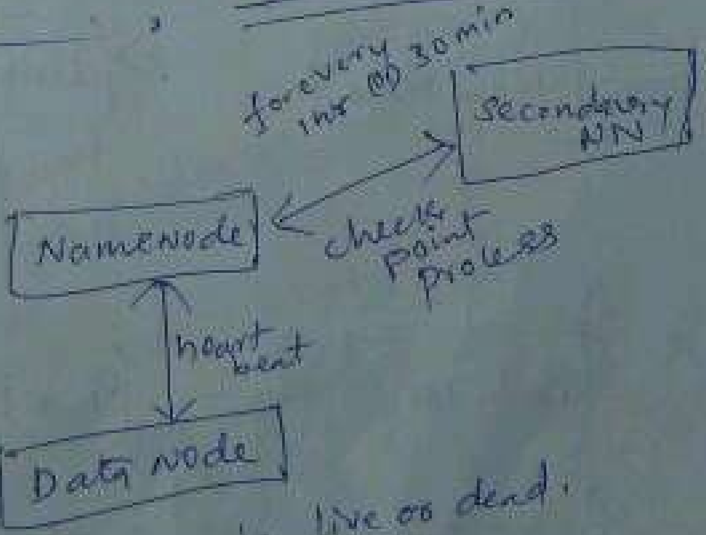
Secondary name node :



⇒ Secondary namnode is housing keeping node, the Communication channel interval between name node and Secondary name node is 30 mins (or) 1hr.

Secondary NN is ~~exat~~ not exact backup for NN.

Communication channel in b/w NN, DN & Secondary NN :



1. for every 3 sec's ie live or dead.
2. 10th heart beat is black report.

Check point process:-

Name node stores meta data in 2 files

i> fs image

ii> edit log files

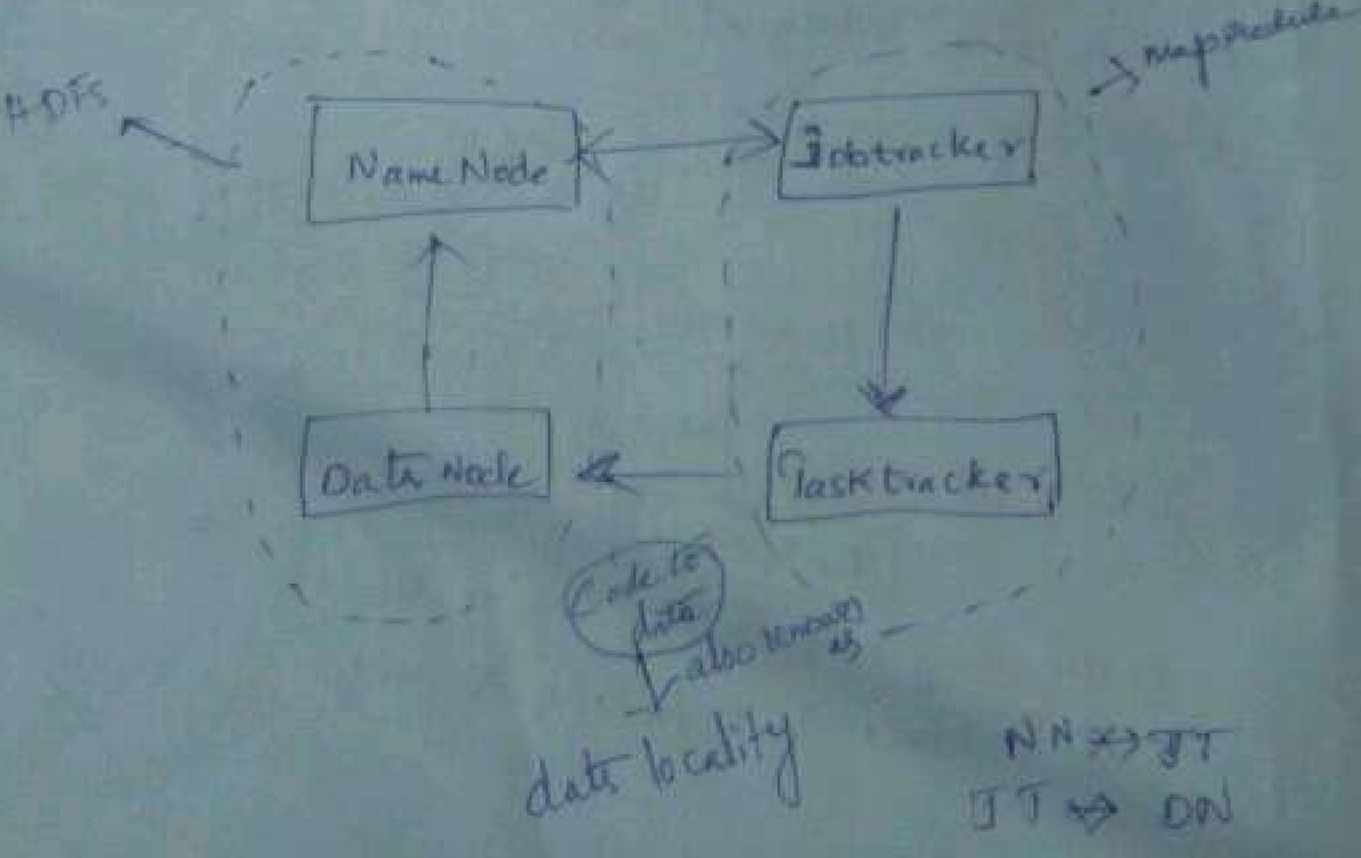
i. fs image:-

fs image is snap shot of your NN metadata.

ii> Edit log files:

Edit log files will update every time when you are storing any data in HDFS.

HDFS vs MapReduce relationship:



N.N <-> JT

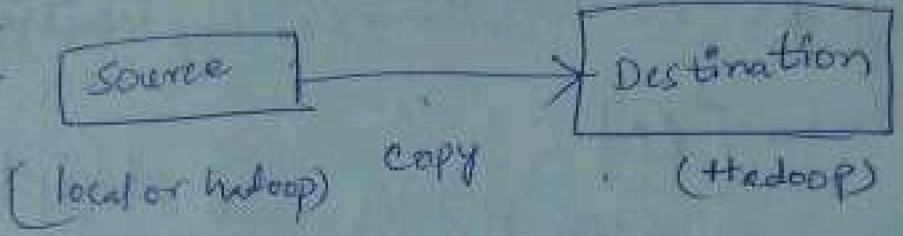
JT <-> DN

16.02.2017

13

Hadoop commands:

PUT

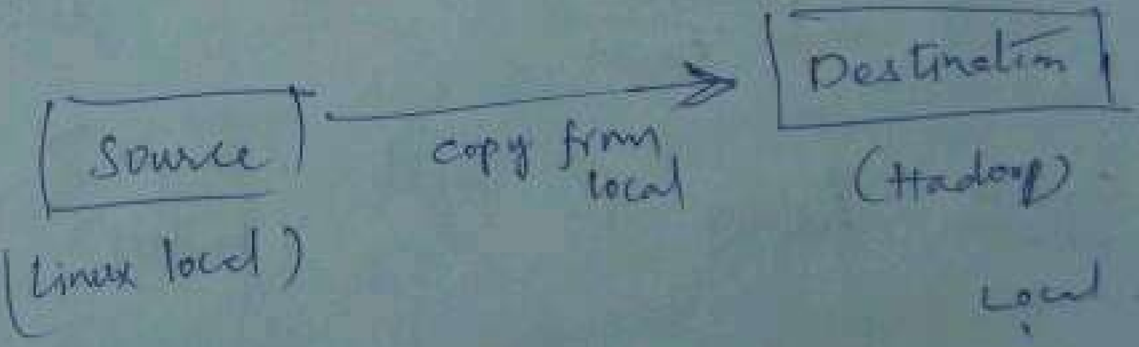


PUT: Syntax:

```
hadoo fs -put <source> <Destination)
              (local or hadoop)   ↓
                                  Hadoop
```

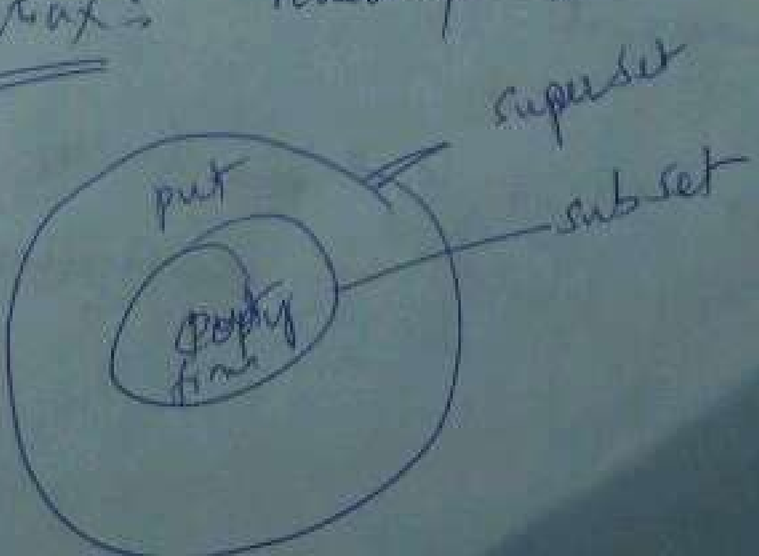
put command can use if you want to copy from any linux server local or any hadoop server one directory to another directory.

Copy from:



Copy from syntax:

```
hadoop fs -copyFromLocal <src>  (Destination)
                    Local         Hadoop.
```



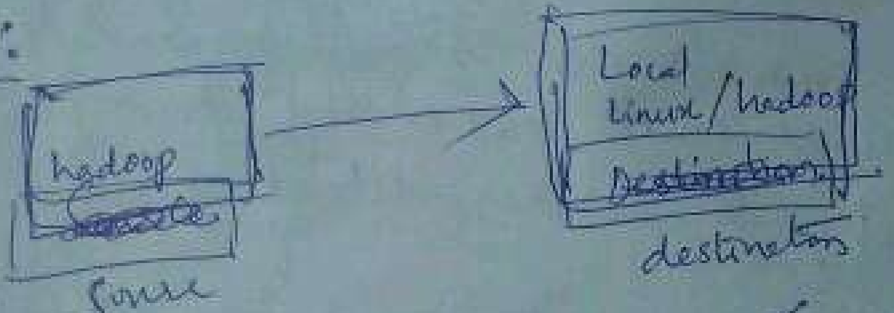
3> ls Command :

Syntax: hadoop fs -ls /user/hive/ → hadoop location

$\boxed{ls}$ → list of files and directories..

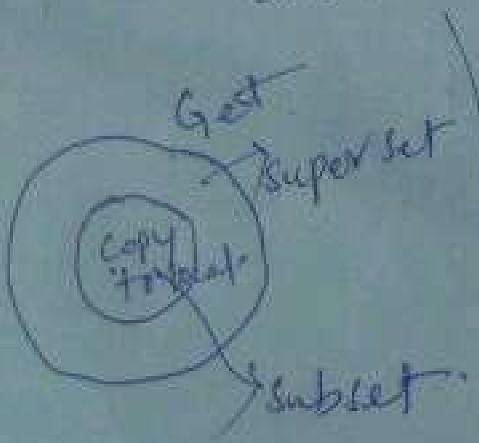
(e). If we want to list files, directories under particular path.

4> get :



Syntax: hadoop fs -get <source> (hadoop) <Destination> linux local/hadoop

(e) If we want to copy data from hadoop environment to your linux local/hadoop environment.



Linux local ⟺ hadoop

hadoop directory ⟺ hadoop another directory copyToLocal (nospace).

5> copy toLocal :

↓

hadoop fs -Copy To Local <Source> <Destination>
↓ hadoop   ↓ Linux local

hadoop location

6> TouchZ:

Syntax :hadoop fs -touchz /user/hive/crcp text.

if we want to create file with 0 byte size by using touchZ command.

7> Cat : If we ~~kean~~ want to see data in a particular file by using cat Command we can see

Syntax : hadoop fs -cat /user/hive/crcp text
hadoop local

Mkdir : hadoop fs -mkdir /user/hive/crcp created.
/user/hive
hadoop fs ~~[illegible]~~ -l's
to check whether file is there or not.

Purge - we can take the delete data from trash

purge table < table name)
→ it will delete entire hadoop

rmdir:-

=> hadoop fs -rmdir /user/hive/crcp
removed crcp

check whether directory is removed or not by using:- hadoop fs -ls /user/hive/.

=> du
chmod
chgrp
chown
expunge } Admin commands (~~[illegible]~~)

owner
group
other
7 7 7
Read-4
Write-2
Execute-1

Linux permissions

7 7 7 → others
Owner group

In our Linux environment

Read=4
Write-2
Exicutable=1

=> As a hadoop developer in Real time we don't have the permissions to change the ACL of particular file or directory.

ACL
(ACCESS control list)

DU : (Disc usage :

Syntax: hadoop fs -du /user/hive)

DU is used to check the files and which are under particular directories size in terms of Bytes.

chmode :

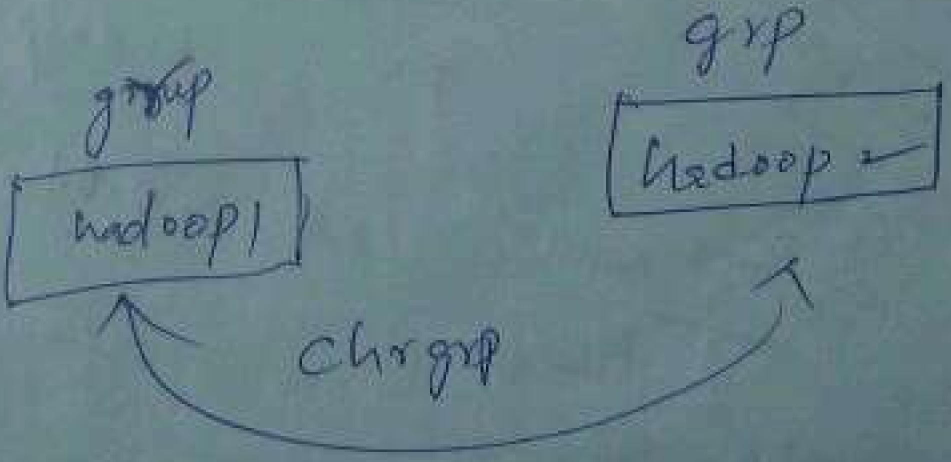
chmod is used to change permissions like Read, write, executable for a particular file(or) directory.

Syntax: hadoop fs -chmod 777 crcp.text.

chgrp :

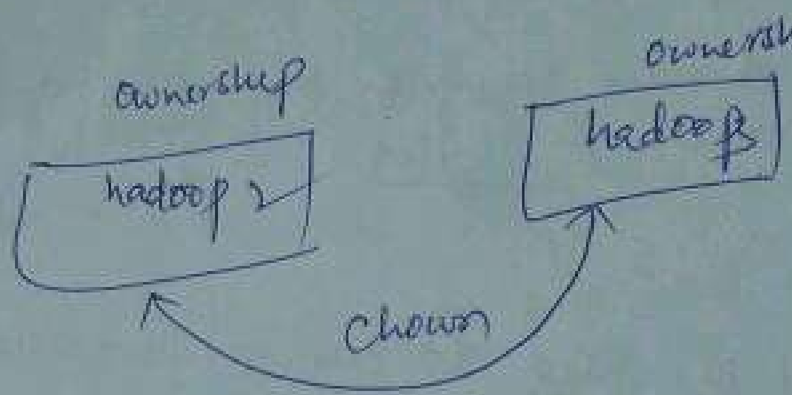
Syntax: hadoop fs -chgrp hadoop1 crcx..

It is used to change the file groups from one group to another group.



Chown:

It ix usedto change ownership of a particular file.



Sytex hadoop fs - chown ~~c~~hadoop 3

Expunge:

Sytax: hadoop fs -expunge

~~hadoop fs -ch~~

=> expunge is used to remove the file from trash

127.0.0.1

2222

~~Usern~~ root

Username : root

pw : hadoop

Host name: 127.0.0.1. port: 2222

Sail :

④ save the machine state.

root : owner

group : hadoop :

Commands: hadoop -fs) for all hadoop commands

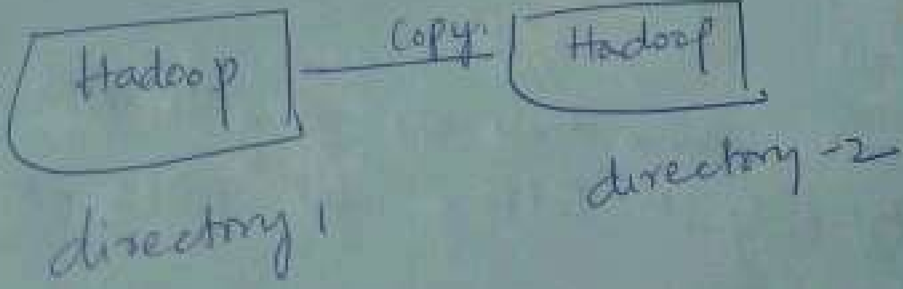
cd- change

ls-ltr-

pwd.

copy and one directory to another directory

Syntax: hadoop fs-cp /user/ destination.



=> cd :/ start-ambari.sh

↓

localhost: 8080

username : admin

pw : admin

**

21-02-2017

65 =

MapReduce

Hadoop configuration files

cd ~~/~~ etc/hadoop/conf
=> spc

1> core-site.xml
Hdfs-site.xml
2> mapreduce-site.xml
3>
4> YARN-site.xml
5> ~~Hadoop-site~~.xml
Hadoop-env.sh

1> Core-site.xml

In core-site.xml ~~Commo~~ they are going to declare Common input/output setting for HDFS & mapreduce

2> HDFS-site.xml

In this they are going to declare properties which are related to name node, data node, Secondary name node & replication factor of our cluster

3> MapReduce-site.xml.

In this xml file they are going to declare properties which are related to job tracker & task tracker.

(iv) Hadoop - env. sh

In this we are going to declare environmentable setting for hadoop.

v) Yarn - site . xml :

In this ~~external~~ xml they are goint to declare properties which are related to resource manager and applications master.

21-02-2017

22

Map Reduce (processing Component)

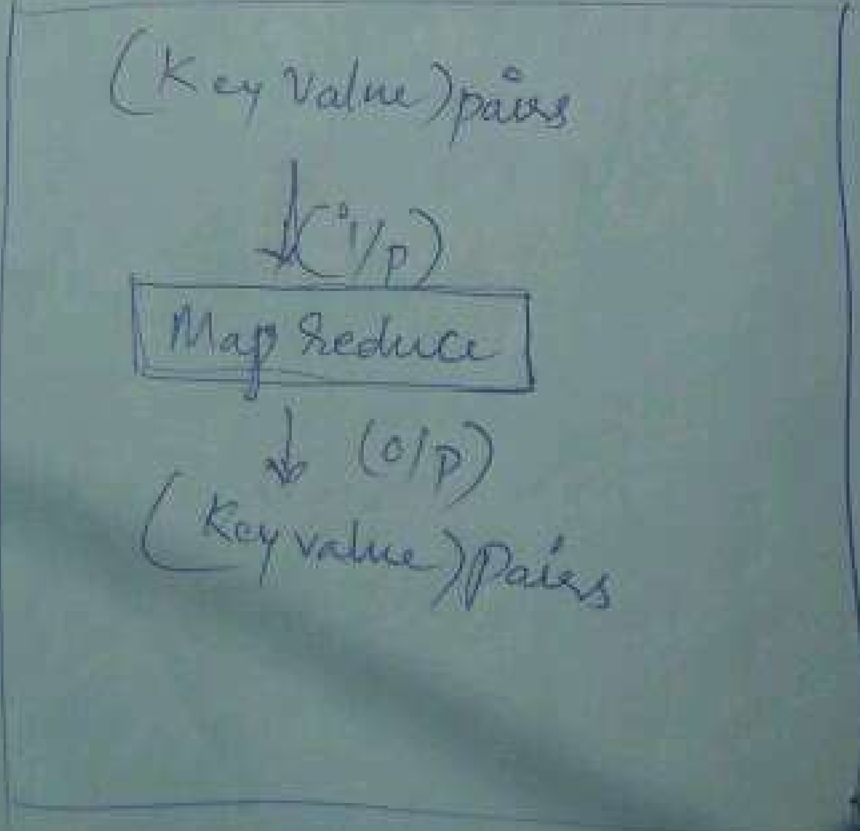
i. It's a parallel processing component.

ii. what ever the data stored in HDFS to if we want to process the data and if it historical data we can process the data by using Map-Reducing frame work.

iii. It supports only batch processing (OLAP)

iv. Map Reduceing frame work totally Really On key value pairs i.e. It will take i/p in terms of keyvalue pairs and it will o/p in terms of key value pairs.

v.



ain class
↓
Driver,
mapper,

Map Reduce Contains 2 Main Components :

i. Job tracker (Master)
ii. Task tracker (Slave)

=> For map reduce we are ~~biting~~ igetting i/p data from HDFS and finally we are storing our output into HDFS.

Steps to debug MapReduce job's

=> When we are submitting mapreducing job (mapreducing, hive, pig), if we want to check the status of particular job we can check by using job tracker url.

=> Job tracker will handle all jobs which are running our hadoop environment.

=> In job tracker url we can check our job details based on job id which is generat by when we are submitted job in hadoop.

(MRV1) (MRV1 version)

i) Main Components in map reduce.

ii) Architecture of hadoop.

iii) Major class in Map reducing.

iv) How map reduce works.

v) Shedulars in our mapreduce

vi) Identity Maper and Identity Reducer.

7. How to monitor mapreducing job and how to get status of our particular job.

8. what are the mandatory steps in Mapreduce

9. How to compose mapreducing job and commands

10. joints in our map reduce

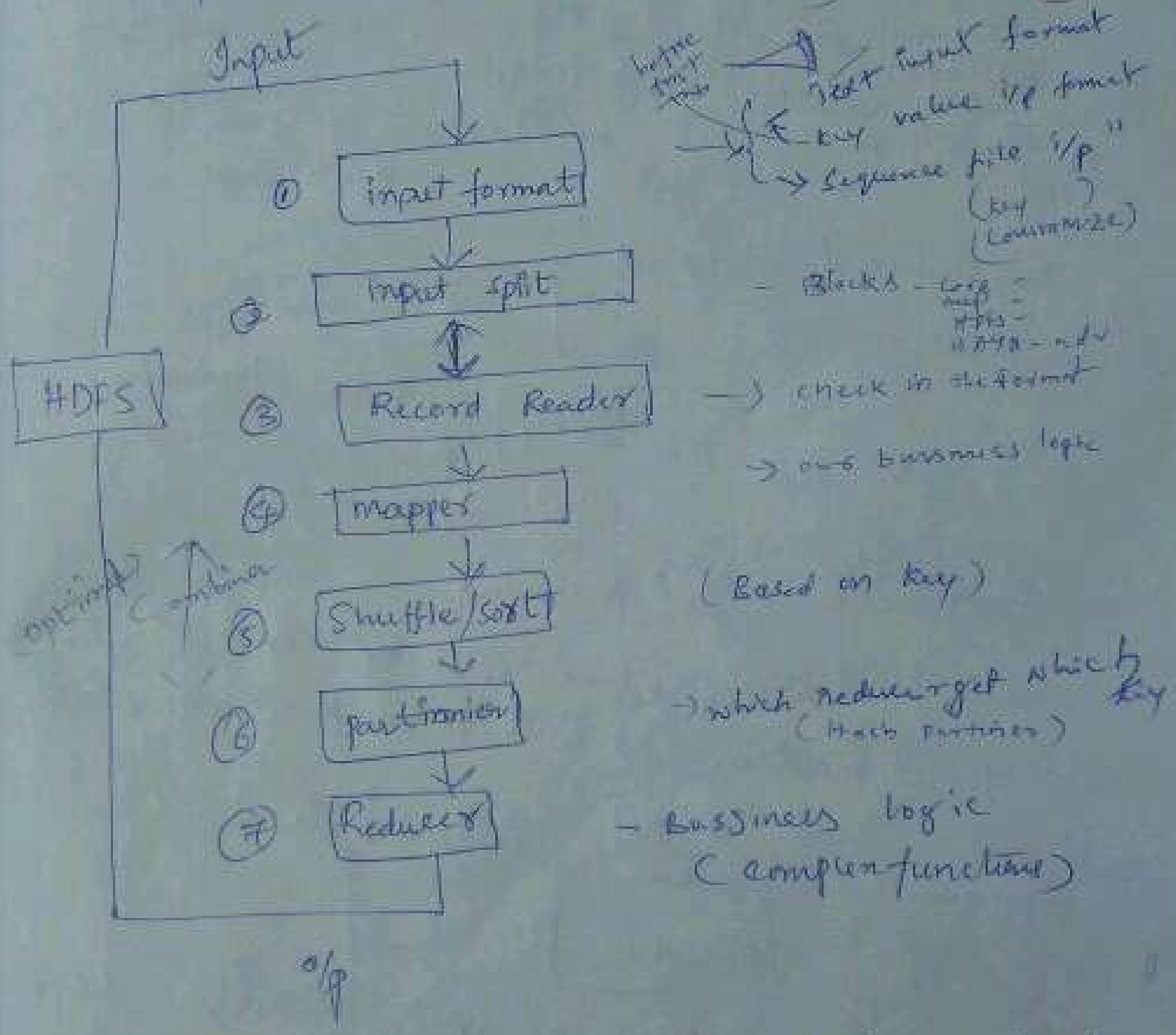
11. ~~Side~~ distribution techniques in MP.

(MR V2)

1. Limitations of MRV
2. YARN architecture
3. How write Map reducing job using YARN
4. What are the main Components in YARN

22.02.17

Map Reduce flow



⇒ Job tracker is to handle the MR Job and at the same time it is going to connect with masters.

23.01.18

1) Textinput → format

Key to offset=key
Remaining=value.

Byte offset value
key | good morning
0 1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 is not tech 20 21
key | Hadoop
Value

2) Key value format:

key | this is crcp
Hadoop | is not technology
key value

3) Sequential file i/p format:

key, value will be constamization
if not mention it will take key atomatically

key
value } Custamization.

Input format

<i> We have 3 types of input formats. 1 option i/p format

1) Text i/p format.
2) Key value input format
3) Sequence file input format
4) Sequence file as a text i/p format

[illegible]

In mapreducing we have 3 main classes.

1) Driver class
2) Mapper class
3) Reducer class.

We are going to declara the our input format in driver class.

Text input format

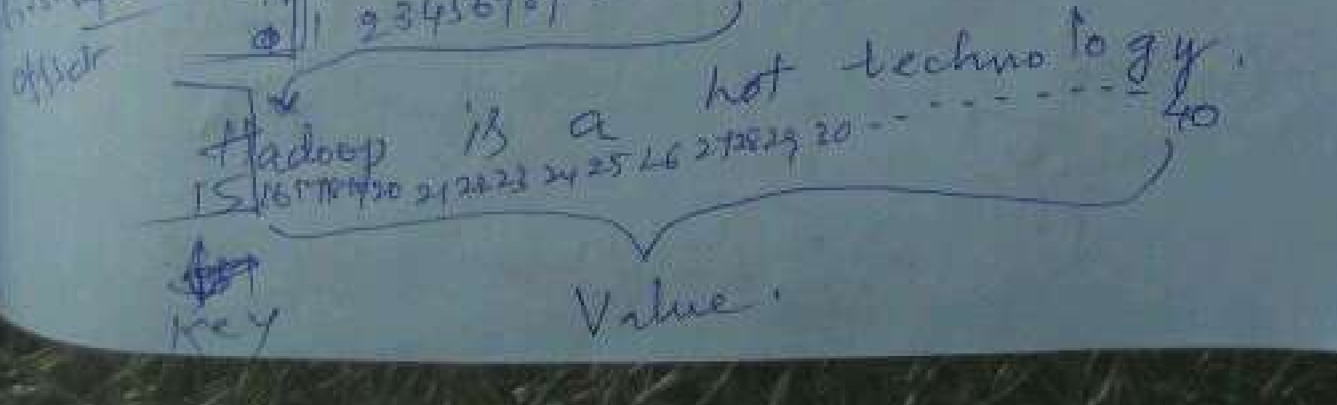
In this format input will be k v's formats of data, so keys and values should be by offset and the remaining information that particular line. particular line

(Key,value)

Byte offset = key

remaining data in a particular line = Value

Eg: Suppose my file contains data like below



<ii> KeyValue input format:

In this key value input format. Key will be before first tab in particular line and value will be remain data in particular line.

Key ⇒ before first tab
Value ⇒ Remaining ~~data~~

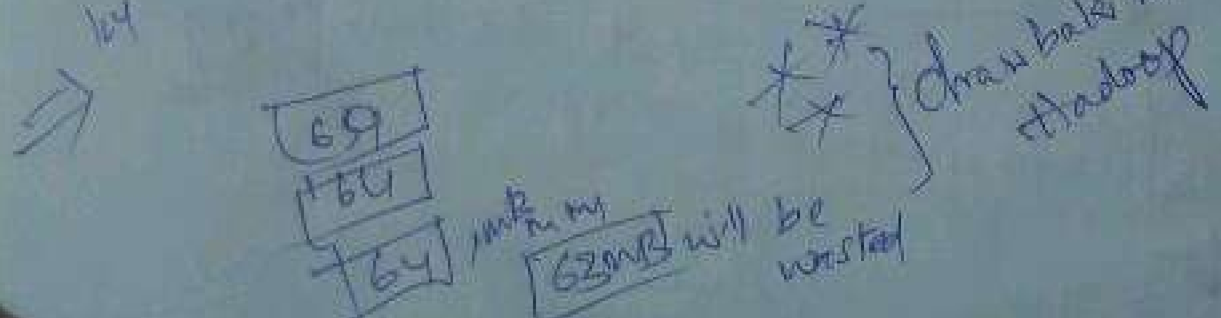
Eg: Hi this is crcp
Hadoop is hot technology

Key — 1st tab — Value

<iii> Sequence file input format:

In this Sequence file input format key, value should be based on client requirement i.e. they can give value of key under and value.

Key } customization.
Value }



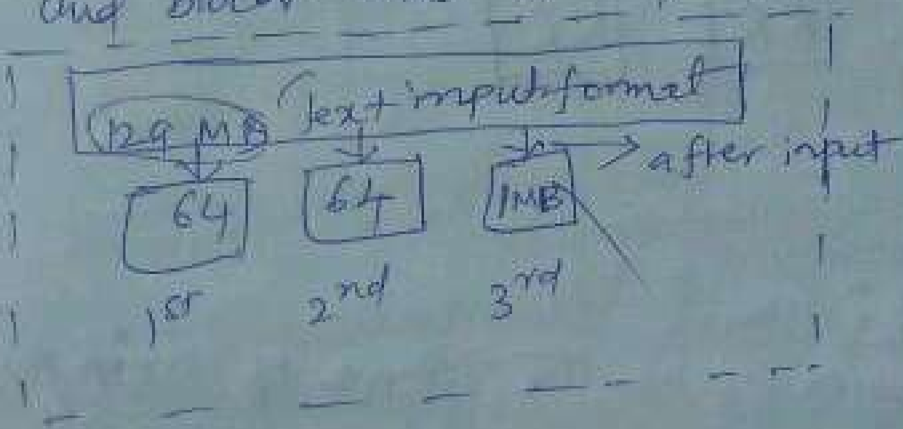
will be wasted

} drawbacks in Hadoop

② Input Split — depends No of mapper

It is going to divide given data into splits based on blocks size.

Ex: suppose is our file size 129 MB and blocks size is 64 MB, we can get 3 blocks

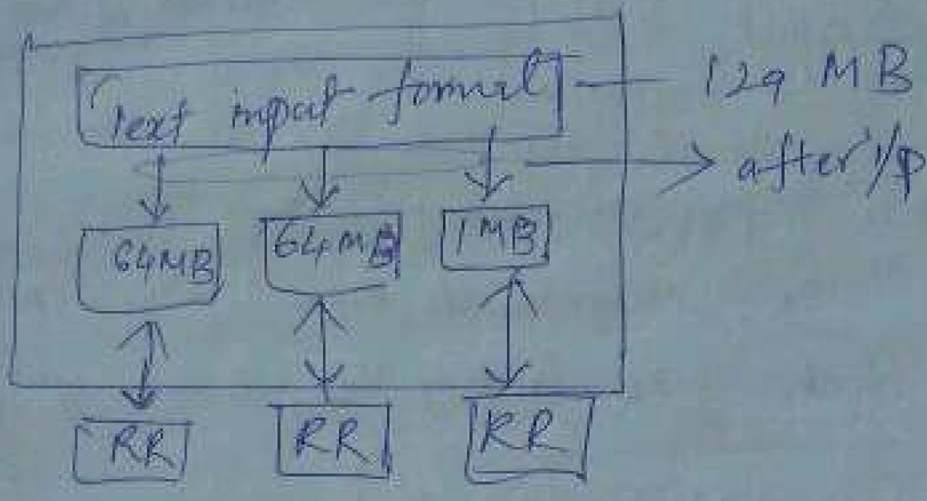


③ Record Reader :—

Record reader is used to check whether the data is in keyValue forms or it is going to convert given data into key value pairs based on input format which are used in the "driver class"

Record reader means it will read each line from each block and convert that into key Value pairs.

NO of Input splits = No of Mappers

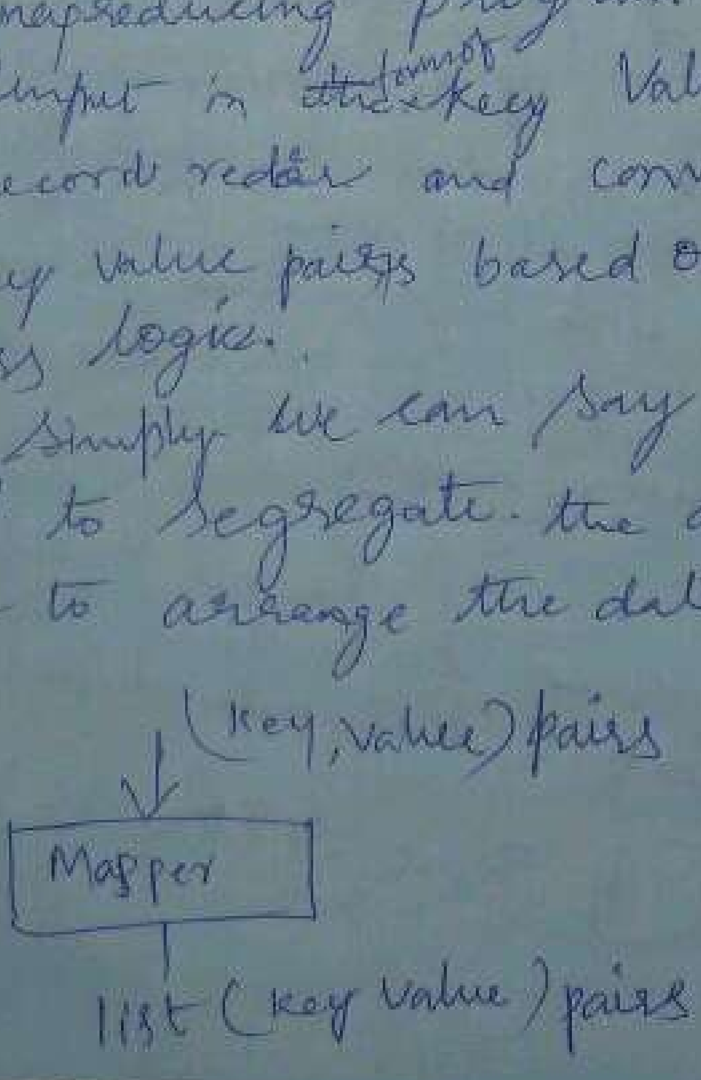


24/02/17

(2) Mapper:

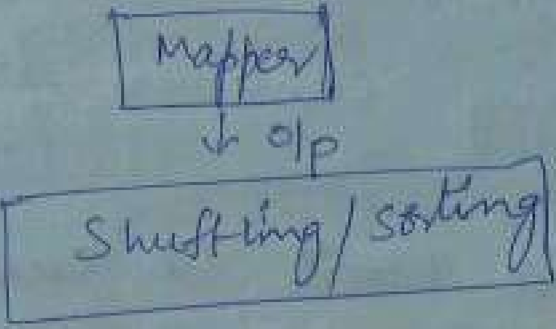
Maper is the ~~type~~ of mainphase in our mapreducing program. It Will take input in ~~the~~ the form of key Value pair from record redar and converts into the key value pairs based on the business logic.

Simply we can say ~~M~~aper is used to segregate the data i.e. we have to arrange the data.



(5) Shuffling/sorting

Shuffling and sorting to arrange the date based on the keys which we are getting from mapper phase.



(6) Partioner: Which reducer should get which key is totally based on partioner.

⇒ Default partioner is Hash partioner. (%mod)

(7) Reducer: Total our business logic will be mapper in our reducer phase. In reducer phase we are taking input from partioner and based on keys which we are getting from partioner simply we can say those many no. of reducers are running in our phase

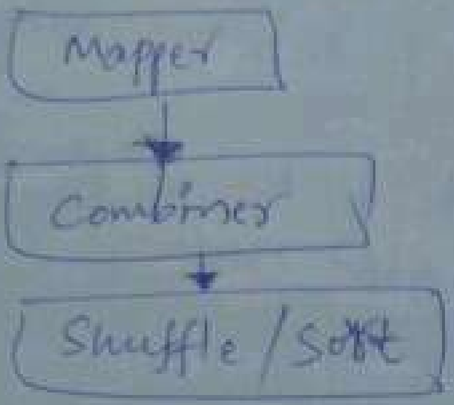
NO of keys = No of reducer

Combiner; (mini Reducer)

combiner is an optional phase in our mapper reduce. combiner is used to reduce the reducer task in MR. we can also call is at mini Reducer.

In mapreduce program we are calling combiner from Reducer phase.

If we want we can implement some Reducer logic in combiner phase also or else we can implement different logic in combiner.



23rd Feb/2017

Input Read

format

Hi ~~Ram~~ This is Ram 64MB

Ram is good guy 64MB

Hadoop is a HT 64MB

Spark is a UCT 64MB

Hi This is Ram

Ram is good guy

Hadoop is a HT

Spark is a UCT

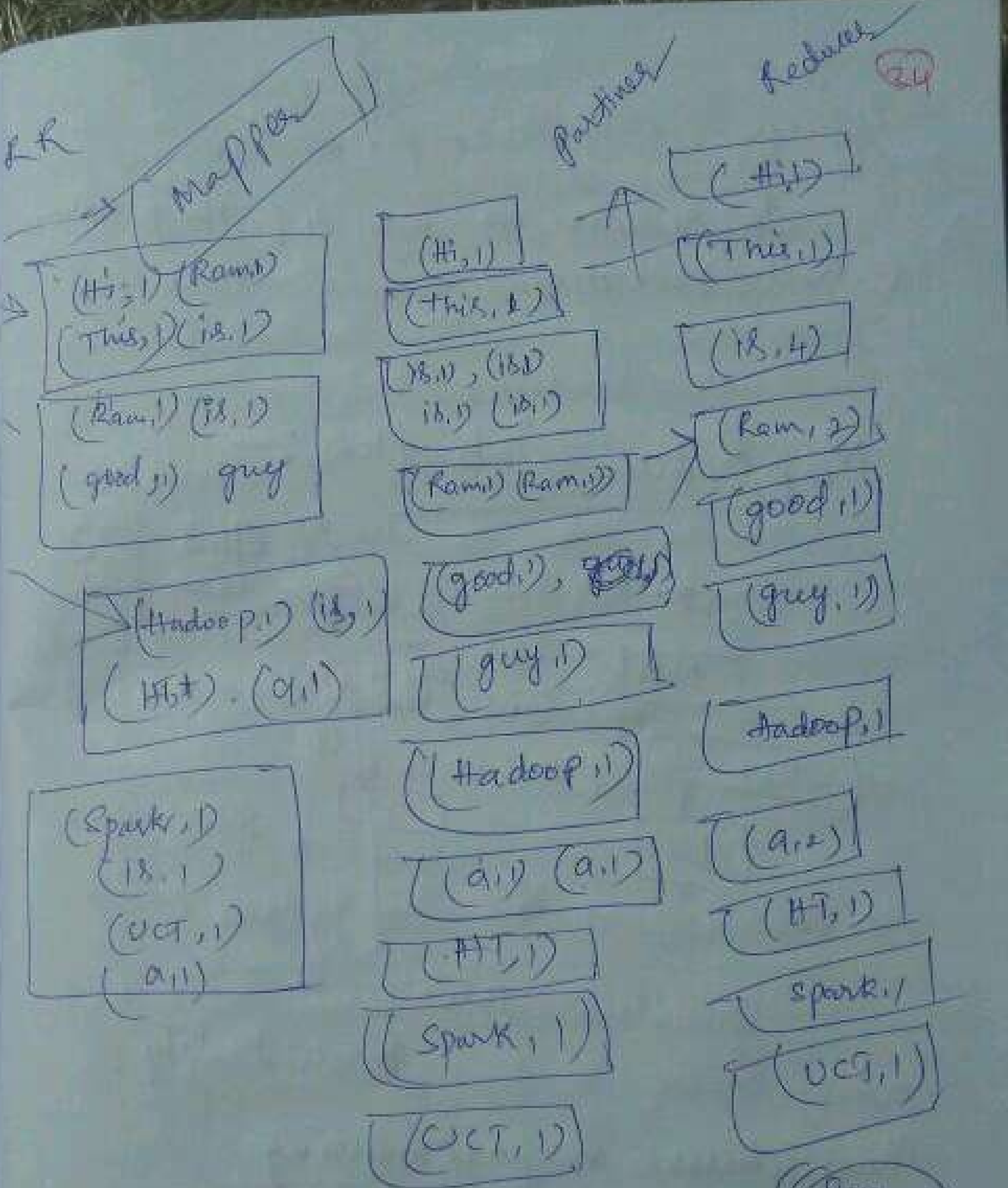
4 split

(Hi, 1)
(This, 1)
(is, 4)
(Ram,2)
(good, 1)
(guy, 1)
(Hadoop, 1)
(a, 2) 1
(HT,1)

(Spark, 1)
(UCT, 1)

15/Feb/2017

plit



In front end → 5 stages

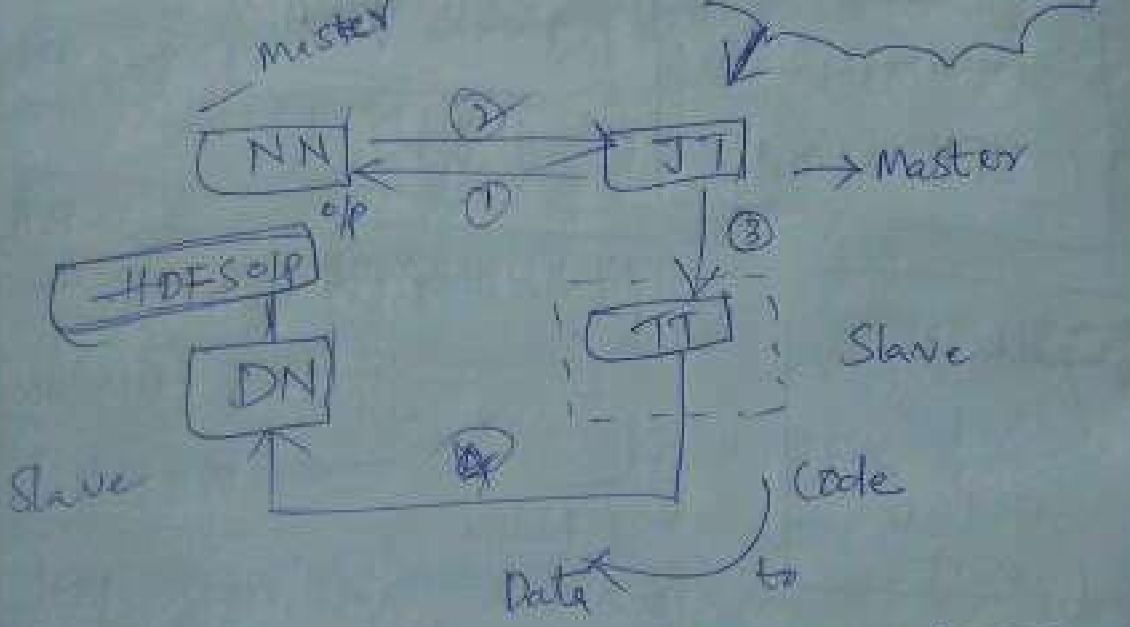
(sum,) function

Mapper = Segrigation (movement of data) from splits
Reducer = aggrilation

Reducer (sum (or) complex~~tions~~ functions) . ✓

=> In reducer we use default partitioner is hash partitioner.

MapReduce job flow :-



Data locality

Main classes in map reducing

1> Mapper class
2 Reducer class
3 Driver class
4 Combiner class
↳ (not a mandatory)

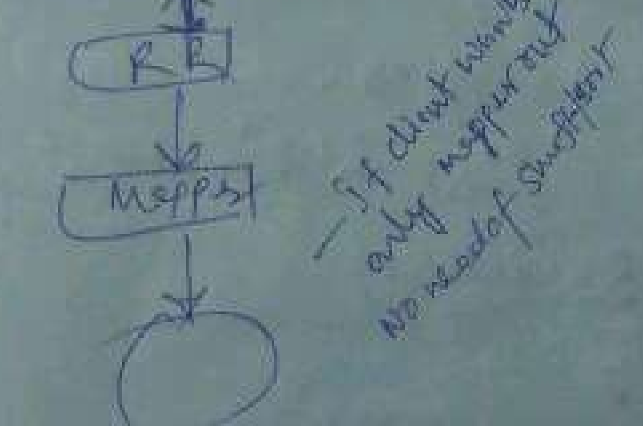
every mapreducing program should have
one or more mapper classes.

⇛ Reducer class is totally based on Bussiness logic i.e bassed on bussiness we can set reducess is equal to Zero.

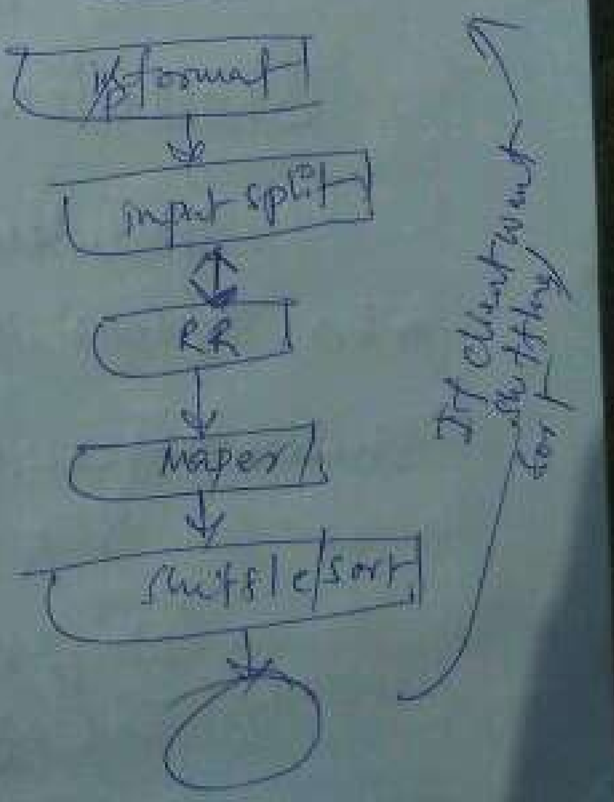
Driver class is the back bone of mapreduce with out driver class we can't run mapreduce.

Combiner class is Optional class based on the size of data and the based on Bussiness logic we can use combiner.

Identity Mapper



Identity Reducer



How to monitor your MR Job and How to get status of your job?

=> For processing in hadoop environment mapreducing is the master. In mapreduce we have 2 trackers (JT & TT)
(Master Slave) ,

In real time client will prove you a jobtracker url, by using that url we can track MR job, pig jobs & hive jobs. why because for any processing job, like, pig job, hive jobs, "in backend" it will run mappreduce.

26.02.2017

Shedulers in Mapreduce:

Shedulers are used to control jobs flow in One Hadoop environment
shedulers are of 3 types.

1) FIFO (first in first out)
2) FAIR Sheduler)
3) Capacity

FIFO:

~~J_1~~ ~~J_2~~

In FIFO Sheduler jobs will run based on admitted time in the cluster that means first job which we submitted in Our cluster it will run first untill to get success message from ~~mapkede~~ (JT). Then only 1st job will terminate independenttly.

In FIFO draw back is suppose the 1st job which we Submitted in the cluster if it is taking ~~to~~ much of time to complete the job. 2nd job will wait untill first job terminate independently.

In FIFO they are maintaing jobs in Pools.

Ex:

J_1	J_2	J_3	J_4

here Job J_1 will Run first

J_2 will wait J_1 Terminate independtly

(pool: collection of jobs.)

Fair: In fair sheduler they are mainteing sustain throssld time for each and every job, based on that time jobs will terminate and reSubmit it into the cluster based on the inprogress Status.

Ex:

Throsold time = 30min for each Job

J1	J2	J3	J4
30min	20min	30m	20mins

here Job J1 will Run first after 30min It will terminate.

J2 will submit after 30 min
J2 will terminate J3 Job will Run.

Capacity Sheduler:

In capacity shedulers they are maintain "Quees" in every pool. based on the job submittion it is going to Runs the jobs in cluster.

* default : sheduler : FIFO **

Mandatory steps in Mapreduce:

1. Job name
2. input format
3. Mapper class
    1> key, value
    2> list (key, value)
    3> for those what are the classes which we are extending.

    for thos what are the interfaces which we are implementing.

4> partitioner: hash partioner
    => Default partitioner => ok
    (or else)
    => customized → which classes we have extended
    → which methods we should use

5> Reducer:

* What are the classes which ~~we~~ we have extende

* what are the interfaces which we have implement.

How to implement customized partitoner?

To implement customized partiner we have to extend "partitoner class" and we have use "get partition method"

getpartition ( )

⇒ Steps to Submitt job in MR environment

I. eclispe	II. Jar (eclipse)
III Hadoop winscp → putty - put	IV Subnit job hadoop jar Jarname - packages. class name HDFS (Soure) HDFS (destion)

27. Feb. 2017

HIVE

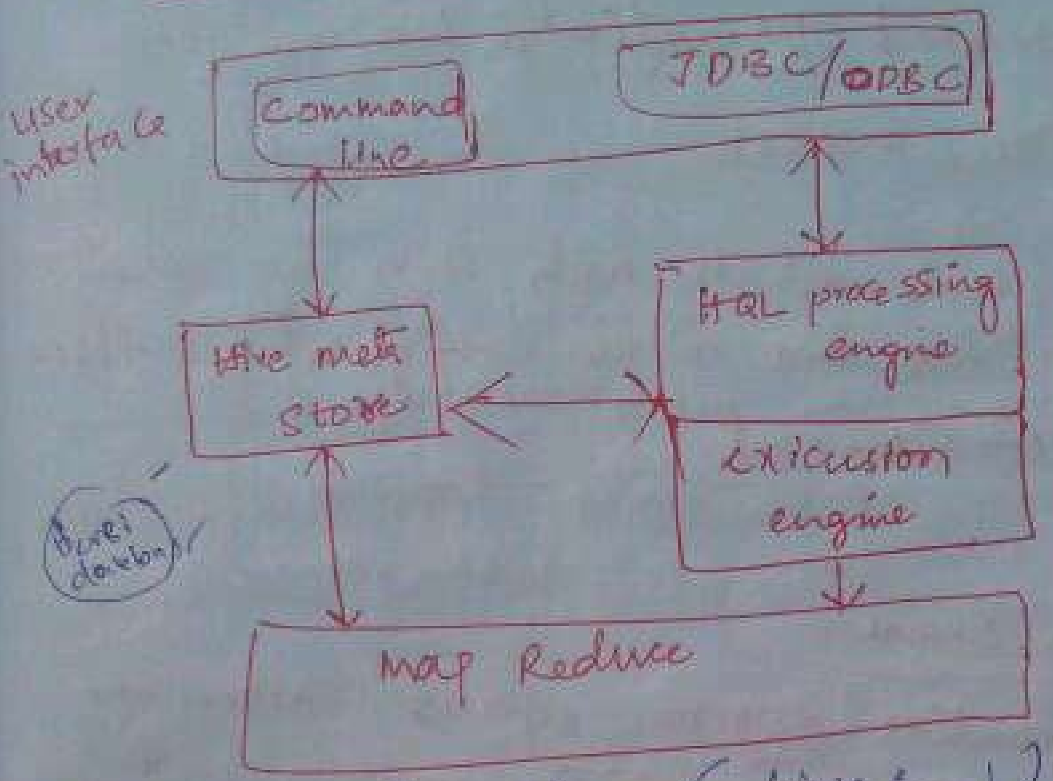
Topics will be covered in Hive.

1> what is Hive?
2> why Hive?
3> Hive architecture
4) Types of tables in hive
5> Types of Joins in hive.
6> Optimization tech in hive.
7> UDFS in hive
8> How to get hive prompt in realtime.
9> What is the use of bealine in hive
& why beeline.
10> How to set temporary properties in hive?
11> Execution engines in hive.
12> parameters substitutions in hive.
13> How to run hive scripts in oozie & production.

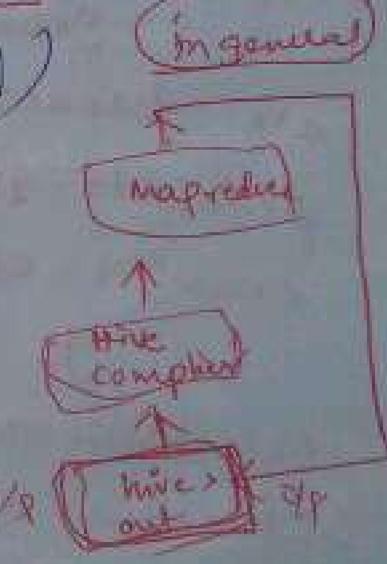
→ SerDE properties
→ Autopurge.

⇒ 14> ACID properties
⇒ 15> Licenses and indxes.
16> modes in hive versions.

HIVE Architecture



To terminate from hive (hive ;↵)



1) we can connect hive in 2 ways majorly:-

i) command line (putty)

ii) ~~IBF~~ JDBC / ODBC

2) when we are submitting any query related stuffs in hive internally 1st it will connect hive query language process Engine (H Compilier). then It will connect exicution engine i.e Mapreduce.

=> In hive we are Storing meta data i.e table related properties in hive meta store

Ex: SQL, MYSQL

we can check which meta store we are using in our hive in the configurations files i.e "hive-site.xml"

Exicution Engine:

Exicution engine means when we are processing any data in environment default it will take mapreducing as a execusion engine.

If you want ~~to~~ check which exicution engine is running under our chester we check by using below command.

Exicution Engine:

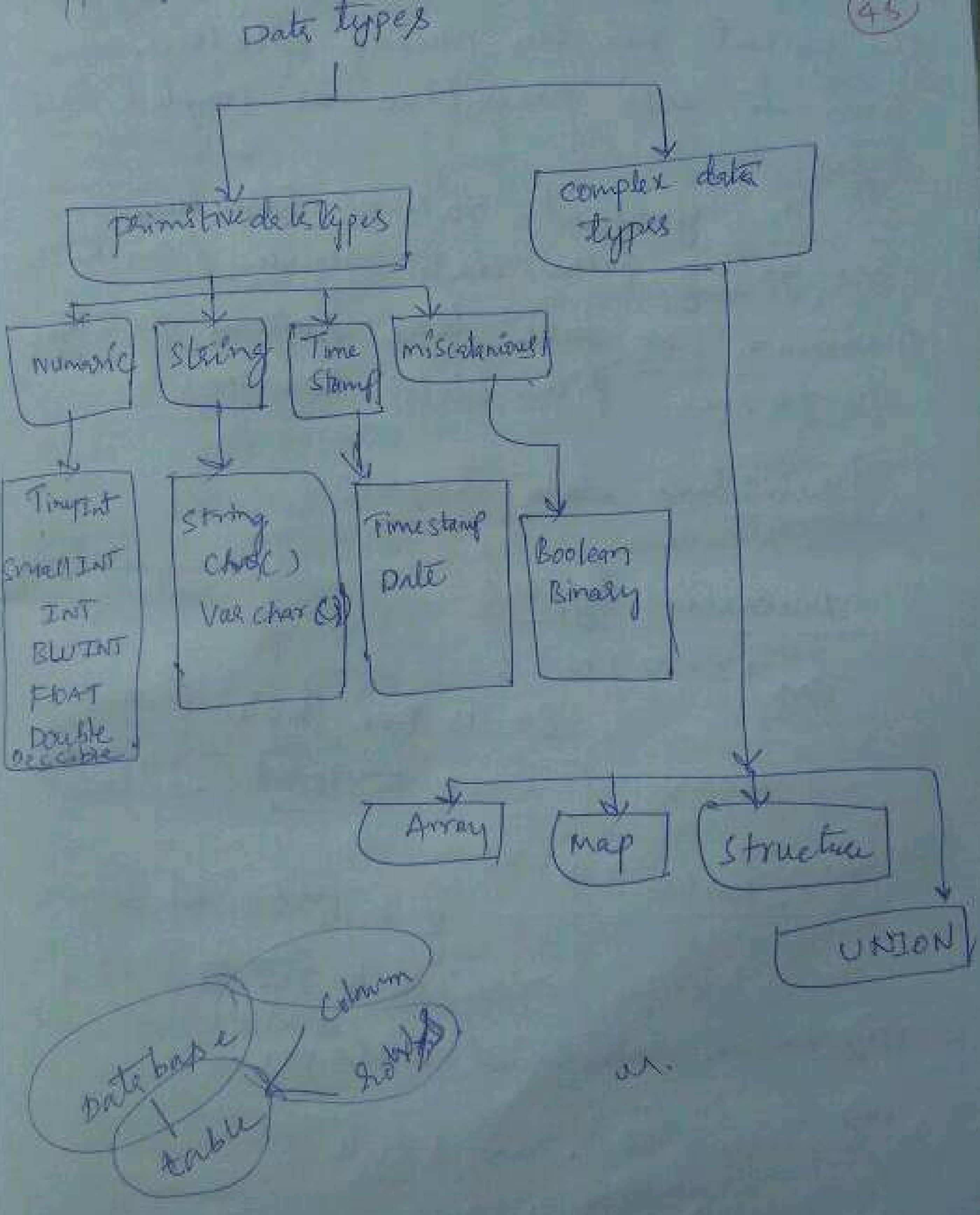
```
hive ↵ enter
hive> set hive.excution.
      engine;↵ enter

MR
```

for ex: it will show: hive.exution.engine = MR
= SPARK

28.02-Feb
2017

45

Types of



ar.

→ Based on the data which we are getting from the client we can choose whether we have to use primitive or complex data types.

In general 90% of structure we are getting from data source from RDBMS. maximum we ~~have~~ can take column data types reference from data source.

primitive data types:

Numeric types:

i) **TINYINT:** It is one byte size integer. $\boxed{-128 \text{ to } 127}$ (-2^7 to 2^7)

ii) **SMALL INT:** 2 byte Singed Integer (−32,768 to 32767)

iii) INT – 4 byte signed Integer

iv) **BIG INT:** – 8 byte

v) FLOAT → (75.86) → 4 byte single precision

Double : (8 byte)

Time stamp:

suppose if we are getting data in terms of date, months, years,

Timestamp : DD:MM:YY : HH:MM:SS : NN

Only for date ⟷ DD:MM:YY

Types in tables in hive:

There are 2 types of tables in hive

i> Internal table (or) Manager

ii> External table

Internal table: (In general time)

If we want to use what ever the data we have in hive Table within hive environment only. It is better to go with hive internal table.

External table:

If we want to use whatever the data stored in hive environment, not only in hive out of hive envirnonment like (pig & MR.

⇒ Internal table is a local (with in hive) external is (global).

Defference's

Internal	External
1> create table emp	1> create external table emp
~~2> drop~~	
2> when weare crating table internal keyword is needed.	1> when (external key word is mandetory)
2> Internal isor local (with in hive)	2> global.
3> When we are dropping internal table It will drop metadata/ as well as data	3> when we are dropping external table it will drop meta data data will be persistance.
4> Locations is not mandatary.	4> Location is mandetory ↓

Default. User/hive/warehouse/.

1.03.2017

Internal table syntax

crcp.csv

Hi,30
Ram,40
crcp,30

(name_marks)

1) Internal table syntax

Create TABLE crcp (name String, mark INT)

Row formated delimited fields
terminated by ','

Location '/user/hive/warehouse'

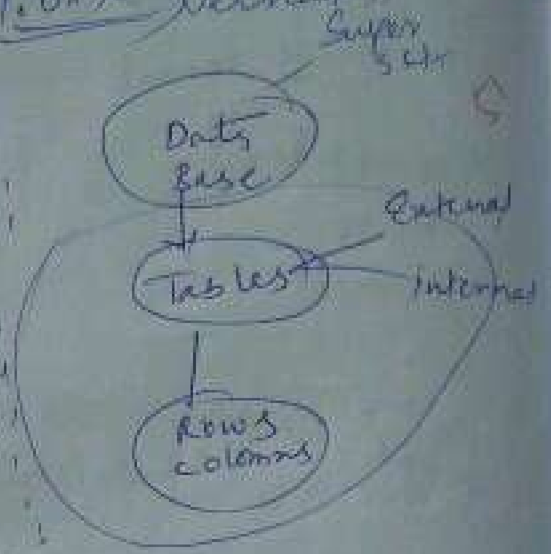
External syntax

2) Create External Table crcp1
(name string, marksint)

Row formated delimited fields
terminated by ','

Location '/temp/crcp1';

Storage / File Formats! (1.0.x) versions



storage formats

1> TEXT FILE → ✗ analytics ✗ performance ✗

[illegible] <-2> Sequence

only analytics } 3> RC (Record colomnar

<-4> ORC (optamization Record coloumnar)

Analytics & performance

Create Table emp (name string)

Row formate delimited fields terminated by ','

Stored as "   "

location '/user/hive/'

Database v/s con

d show tables

Create database D_1 ;

use D_1; → emp1
Emp2

Create data base D_2;

use D_2; → emp2
– emp3

~~Cred~~
show data base D_1; show tables

emp1
Emp2
Create emp1;
already exist;

default data base : default

for 0.1.x Versions

Storge formats are of 4 types:-

1) TEXT FILE

2) Sequencefile

3) RC (Record coloumnar) / or Row colomnar — 2.x Vrs

4) ORC (optimized Record colomnar / or optamized Row colomnar

TEXTFILE If we are not doing any analysis
and if we are not perfomance as
a major factor and if Wee are
getting date with comma, & tabe
(,/) seperated so better go with

TEXTFILE — ✗ Anlysis
— ✗ perfomance

9,/

Sequencefile:

If we are not doing any analytical things and we are not concentrating on performance. if we are getting data with fields terminaters (;,|) with less file size less than (<64MB) better to go with Sequencefile.

Sequencefile → ~~analytics~~

→ ~~performance~~

→ less than 64MB ✓

(RC) Row colomnar :- → ~~Analytics~~ ~~performance~~

→ only analytical

If we are doing any analytical kind of things and we are not taking performance as a major factor.

Row colomnar → (Analytics ✓)

Optimized Row colomnar: (ORC)

If we are doing Analytical things and considering performance as a major factor.

Task: ① Create internal table x external table

② check which exicution is running default

③ How to change the default execusion engine.

Textfile

```
Create external table crcp (name, string)
Row format delimited fields terminated by
','
stored as 'Textfile'
Location '/temp/crcp';
```

Sequencefile:

```
Create External table crcp1 (name string)
Row format Delimited fields terminated
by ','
Stored as Sequencefile
location '/user/hive/crcp1';
```

Rc file:

Create external table crcp2 (name, string)
Row format delimited fields terminated
by ','
stored as Rcfile.
Location '/user/temp/crcp2'

ORC file

create external table crcp3 (name, string)
Row format delimited fields terminated by
';'.
stored as ORC
location '/temp/crcp3'

To check colomns with & to data types of a particular table

syntax: describe crpe;

To check end to end properties of particular table

syntax: Describe crcp;

=> Describe extented crcp;

Exicution Engine in hive :-

Exicution engines are 3 types

i. mapreduce

ii. Tez

iii. Spark

```
hive> set hive.execution.engine = Tez;

set hive.execution.engine; ↵
(To check)
```

What ever the setting we are doing under hive prompt i.e only for temporary purpose.

If we want to make it as a perminant admin will add that properties in "hive-site.xml."

How to set tempory ~~perp~~ properties in hive?

By using set command we can set temporary properties in hive.

i.e

Syntax: Set hive. executionengine = Tez;

=> Beline usage in real time:

we are using Beline to provide Authentication & Authorization (security) to the hive server.

for beeline we need 3 properties:

> Connection string endswith 10,000

2) user name:

3) paswad :

How to run hive scipts in Oziee

If we want to run hive scipts in Oziee or production environment we should save that hive ddl or DML's with extention ".hQL".

eg: hive -f .hql

when we are running .hql mannaly we can store it in linux local & hadoop

when we are running through Oziee (shreduling) we ~~can~~ should place .hql file in hadoop local.

03-03-07

How to load date into hive tables

* we can load date into hive tables by using below commands

Syntax: If we have data in our linux environment. we can load data into hive table by using below comman

— linux

Syntax: Load data local inpath '<path>'
into Table CRCP;

* If we have data in our hadoop environment we can load data into hive table by using below command.

Syntax: Load data inpath '<path>' into table CRCP;
↓
hadoop path

If we want to "check" data in our hive table we can use below command

Sy: Select * from table name;

If we want to check the metadata of our table we can use

Sy: Describe CRCP;

How to daad data into final table By filtering data from stage table

Appending

Syntel

```
Insert into table CRCP_files
        Select name_id from crcp
where id >500';
```

Condition! Stage CRLP => name_id
=> id> 500

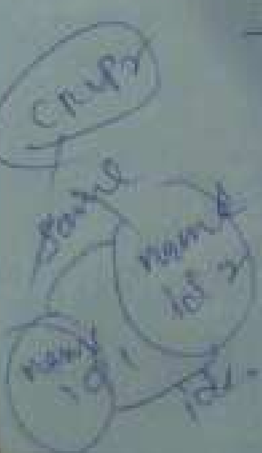
final -> crcpfiles

update:

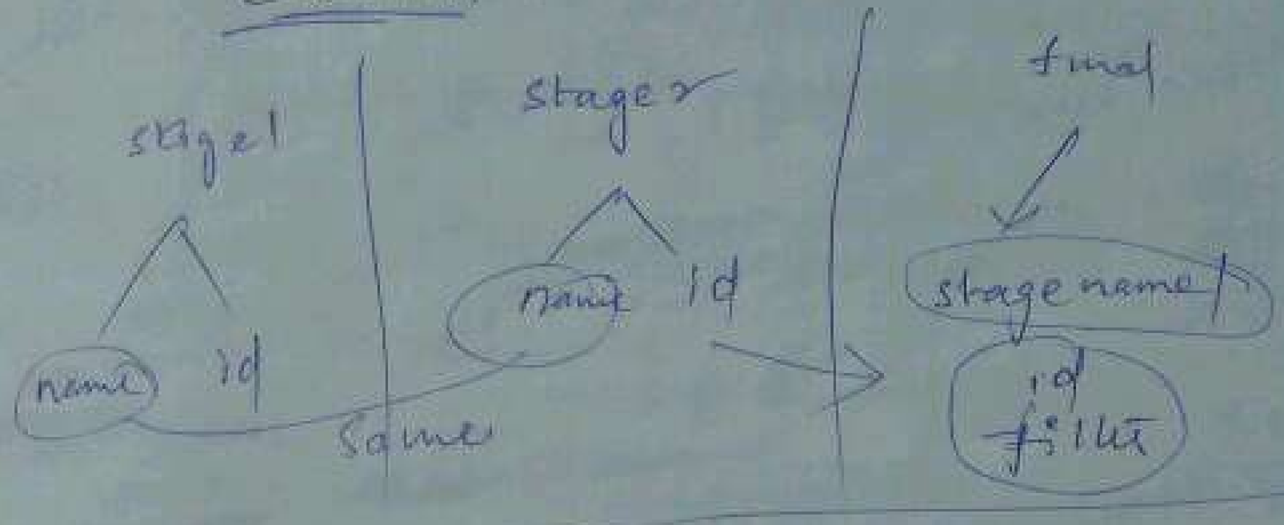
```
Insert overwrite Table CRCP_final
    -> Select name, id
~~select~~ CRCP where id >500';
from
```

=>

```
Insbfut overwrite id_crcpfinal
    Select name_id from crpcl
from CRLP where stage/name = stage2 name
```



Condition :



Syntax :

```
Insert into Table CRCP3
select t1.name from
stage t1 JOIN stage t2 ON t1.id = t2.id
```

Joins in hive :

- JOIN
- LEFT OUTER JOIN
- Right OUTER JOIN
- FULL OUTERJOIN

Create database HIVE;

~~Create tab~~

use HIVE;

Create table order1 (id int, mrp int,
product varchar (10))

cm

row format delimited fields terminated by
','

Stored as TextFile;

~~location '/tmp';~~

Location: describe formatted : table name

name : /appx /

=> hadoop fs -ls /apps/

we want to run hadoop commands in hive;

Sy:- hadoop fs -ls /tmp/ ;

with out entering hive:

hadoop -e 'select a from TableA' ;

hadoop fs -ls '/' ( )

name

Optimisation techniques.

i) CBO: Cost Based optimisation

CBO is nothing but when we are doing "join" kind of things we have to check count of rows in 2 tables.

After we have to compare less no. of rows with more no of rows tables.

Ex:

A
50 row

B
1 row

JOIN

```
Select t1.Name from
A JOIN ON A.id = B.id
```

↓ More rows ↓ Less rows

ii) Vectorization:

Vectorization concept is used to increase the performance by applying vectorization features. (Vectorized)

Property: 1) Set hive.vectorization.execution = true;

2) Set hive.vectorization.execution.enabled = true;

object row

instead of comparing each and every ~~[illegible]~~ . sequencilly by using vectorization we can compare bulk rows (1024) at a time parellely.

If we want to enable vectorization features in our hive prompt (environment) we have enable the 2 properties

Syntax:

```
Set hive . vectorized . execution . enabled
                                   = true;
```

(iii) partitioner!

partitioner is used to get better performance and at the same time cluster running time will be less. (hive server)

Partitioners are of 2 types

1) static partition

2) Dynamic

⇒ Static partition means we are passing static information (partition) manually.

⇒ In dynamic partition we are not passing partition Values manually It will take partition values dinamically from the table.

Default Static partition will be enabled, But we have to enable ^ manually.

dinamic partition

```
Set hive . exec . dynamic. partition = true;
Set hive . exec . dynamic partition . mode = strict;
set hive exec . dynamicpartition.mode = non strick;
```

Static:

```
Create external table emp;
(name string, id int)
patitioned by (static string)
Row formated delimited fields terminated by ','
Stored as ORC
```

Dynamic :

Name	id	State

Sytax :- Create external table emp.

(Name string, id int)

partitimed by (state string)

Row formated delimited fields terminated by ','

Stored as ORC

Location '/tmp/ hive/emp';

How to load data into dynamic partition table :

Table A

Name	id	State

Stage table without partition

txt-file

Table B

Name	id	State

with partition (Dynamic)

load data from table A to B.

Suppose table A has 3 colomns i.e

id	State	name

i.e = Stage table

without partition with text file format.

and tabe B has 3 colomns i.e named State id

(repartiong table without partition with dynamic)

then i want to load data from table A to B

Table A to Table B

```
Insert into table Table B
partition (State)
select Name, id .State
from table A;
```

```
Insert into table Table B
partition (id:)
select id, State, id
from Table A;
```

suppose table A has 3 colomns i.e

id	State	name

i.e = Stage table

without partition with~~g~~ tex+ file format.

and tabe B has 3 colomns i.e named State, id

(repotiong table without partition with dynamic)

then i want to load data from table A to B

Table A to Table B

```
Insert into table Table B
partition (State)
select Name, id, State
from table A;
```

```
Insert into table Table B
partition (id!)
select id, State, id
from Table A;
```

when we loading data into partition table from unpartition table. In selection statement partition column should be last column in the selection statement

~~at~~ a [illegible]

→ DC

```
Insert into table Table B
partition (state)
Selection Name, id state
from tableA;
```

└── N.

How to load data into Static partition: or local directly from Location

Syntax:

```
Load data local inpath '<>'
into table tablA
partition (state='KA');
```

from one table to table

```
Insert into Table table B
partition (State = "KA")
State Name, id,
from table A
~~[illegible]~~ where State = "KA"
```

Dynamically pick values

create the rule condition

⇒ To load data to static partition
 i) table to table
 ii) table to location

Configured load; partition

when we are using partition based on the partition column values internally it will create ~~[illegible]~~ Sub directories. based on unique values.

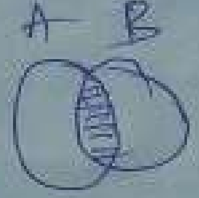
Drawback :-

Suppose if we are using primary key as partition ~~column~~ column in our table internally it will create Sub directories based on primary keys (unique).

JOINS

1) JOIN
11) LEFT OUTER JOIN
3) RIGHT OUTER JOIN
4) FULL OUTER JOIN

1) JOIN:



primary key : 1
Composite key : more

Table A — Table B

Name (id) — Name (id)

(primary key)

If we want to get common recods ~~in~~ ~~the~~ b/w two tables we ~~have~~ can filter based on primary key, composite key (collection of colmns)

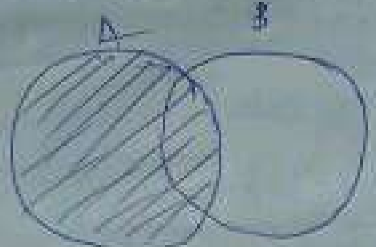
Ex: Table A has two colomns i.e Name, id.
here id is primary key.
Table B has 2 colomns i.e marks, id.
here id is primary key.
If I want to join table A and B I should use Id has primary key.

P.K → primary common = id

JOIN condition:

```
Select Name, marks
from TableA JOIN TableB
ON TableA.id = TableB id.
```

ii) LEFT OUTER JOIN:

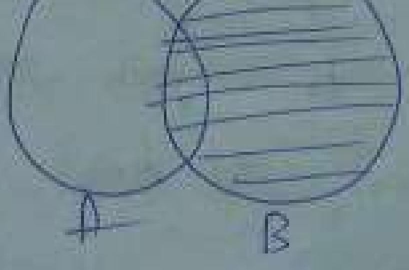


Left out join means if we want to get common rows in b/w 2 tables besides which are not common in left table in join condition

Syntax:

```
select Name, marks
from Table A LEFT OUTER JOIN TablesB
ON TableA.id = Table B id.
```

iii) RIGH OUTER JOIN



Syntax: Select Name, marks

from Table A RIGHT OUTER JOIN TABLE B
A

ON TABLEA.ID = TABLEB.Id

Table A

Name	Id
H	1
F	4

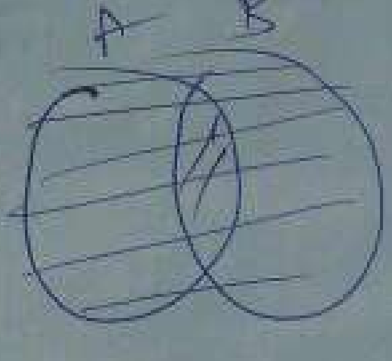
Table B

Salary	Id
4	1
10	10

Right outer

Name	Salary
H	4
NULL	10

iv) FULL OUTER JOIN



Select Name, Salary

from Table. FULL OUTER JOIN TablesB

ON Table A.Id = Table B.Id.

Full outer join o/p

Name	Salery
H	~~4~~
F	NULL
NULL	10

~~[illegible]~~ Use case: → Insert Into

Table A

id	Name
1	H
2	M

Table B

id	Name
1	H
3	R

→ o/p

id	Name
1	H
2	M
3	R

Syntax :-

select Name, id
~~from~~ ~~table A~~ insert into

* * Views:

views are nothing but logical layout ~~sent~~ of data i.e if we create views one time based on some conditions on top of base table view will be updated automatically when we are getting based table.

Syntax:

```
Create view CRCP as
Select * from Table A;
```

100

ORDERBY / SORTBY

order by: Orderby means if we want to arrange given data in a certain format based on the colomn.

Sort by: If we want to sort the data based on the colomns.

7.3.2016

partitioned by – ' (high cordnalty i.e = [illegible])

Bucketting:

- Bucketing is used to overcome the draw back of partitiong i.e (if we are using the column which has high cordinality internally it will create sub directories on based on cordinality.

Suppose if the column has On lakhs distint value. internally it will create more one lakh sub directories.

To overcome that they came up with Bucketing.

→ Bucketing is used if we have a column with high cordinality (no. of unique occurance) with equal distrubution.

we can mpement bukettng by using keyword "clustered by"

```
create Table emp (emid Int, state string)
cluster By (state) IN 30 ~~Block~~ vckets.
Row format delimmies
      files Terminated by ','
stored as ORC;
```

If we want to enable bukketing feature we have to use hive. enforce equal ched bucking.

```
set hive . enforce . bucketing = True;
```

How to Load data into bucketing table

Syntax: Insert into table emp
    select * from emp;

→ Emp →state
    insert into Table emp
    partition (state)
    select id, state from emp;

partitioning	Bucketting.
1) we can apply partitioning technique in one or more colomns.	we can't apply Bucketting in more than one colomn
2) If partitioning internally it will creates sub directories.	2) Internally ~~Iff~~ it will create files.
3) If we have less cordnality we can ~~prefer~~ partitioning ~~dir~~	Cordnality for 4) If we have high cordinelity for the particular ~~[illegible]~~ distruction with equal can be bucketing.
4) We can impletiment this by using key partitimed by	4) clusteed by.

Internal process of Bucketting.

When we are using bucketting concept internally it will follow (hash mod n)

(h mod f)

n= buckets
h=value of colomn.

Table Sample:-

Table sample is used to check the sample data from the bucketting data.

we can apply by using below table sample

syntax:(keyword)

Tablesample (bucket x out of y)

y - stands are values in each group.

x - stands for colomn value data in the bucketing table.

1) Create Table emp (empid, int, state string)

=> ~~Nor~~cal

clustered by (state) INTO 32 Buckets

should as ORC;

II Set hive . enforce . bucketing = True;

III Insert into table emp
Select a from emmp,

IV . state * from emp
(Table sample (Bucket 6 out of 8))

mapjoins :

map join is used to give small size table information to the compiler. so, that we can increas aeto

We can implee Map join keyword with map join .

Synt: ex

Table A	Table B
1 lakh	10 Records /

<u>mapjoin syntax</u>:

small size table

```
Select /*+ MapJOIN (TableB) */ *
   from table A JOIN Tables ON
      TableA.id = TableB=id;
```

huge table

```
Select /*+ /*+ streamtable (TableA) */
                       * from;
```

hints: giving hints to the compiler (Hi

1) parameter . Substitution:-

We can pass parameters in hive by using Keyword set and we can call parameter value by using "hiveconf'

ex:

i) SET current_date = '2017-03-19';

↓ (NO Space) ✗

${hiveconf : current_date}

ii) how to call parameters in hive

set current date = '2017-03-09';

Select * from emp
where Time ≤ ${hiveconf :current_date}

Ex: 2

Set name = 'CRCP';

Select * from
emp where name != ${hiveconf :name}

Sortby (Local)	order by (global)	~~distributed~~by Distribute by	Clusterby = Distributed by+Sortby
1) It is a local method of apply on logic on whole data we are applying some arrangement technique locally on each reducer 3) we can use sort by technique "sort by"	1) order by is a global. 2) we are applying this technique on top of bulk data, but in real time there may be less chances to use this one. 3) we can use orderby technique by using "orderby" key word.	1) It is nothing but if we want to distribute rows in b/w reducers by we can distribute by using "distributeby" keyword.	It is combination of distributedby and Sortby.

Task: How to findout Distrint Values without using distrint key

Avro's - less data
ORC: more data

Advanced storage formats:

1) AVRO → (all colowms)

2) parquet → colowmn

① ⇒ Avro's used to pick all rows in all colowmns (row based). simply we can call avro's (avro row based format)

② ⇒ parquet is a colowmn oriented storage format, i.e if we want to get specific colowmn values we can use parquet.

Syntex :⇒

```
create table emp (name String)
row formate delimited fields
        terminated by ','
stored as avro
Location '/temp';
```

Syntax: create table emp (name string)
row format delimited fields
terminated by ','
Stored as parquet
location '/temp';

UDF : User define functions

User define functions means if we are not able to satisfy the requirements by using inbuilt operators and functions in hive we can achive by using UDF

UDF's are of three type

i/p	o/p
1	1
Many	1
1	many

i> UDF

ii> user define aggrigative function (UDAF). Ex: ~~term~~

iii> User define tabular function Ex:

(UDTF).

↓

Ex: {a, b, c, b} = a
b
c
d

	i/p	o/p	exmple
UDF	1	1	trim
UDAF	many	1	Avg
UDTF	1	many	array

{abcd} =>
a
b
c
d

** Windowing function
**
- → Rank
- → Dese
- → rownum

Jack Ankit How to findout distinct values without using Distinct key.

```
SELECT employee_location from emp
Group BY employee_location
```

emp empple	emp locatn
A	Neyork
B	India
C	Rusia
D	Neywork
B	India

Nuywork
India
Rusia

9.3.2017

'Tamplets to implement UDF's in hive functions 81

I) eclipse
code implementation

II
Create a Jar
in eclipse

III
Add Jar in our hive
environment

```
hive >add Jar <jarpath>;
```

IV
create tempory
function trim as
'Org.apache.trim';
select trim(colomn)
from table';

```
hive> list -jars';
```

tempory function

Windowing functions: * * * *

① Row number ② Rank ③ Dense Rank

Table a

Name	
a	a
b	b
c	c
a	a
a	d
e	e
a	a
f	f

=>

Name	
a	1
a	2
a	3
b	4
c	5
e	6
f	7

Row_number ~~s~~ o/p

=) Imp ~~x~~
Syntax: How many ~~Red~~ reducers are using in

```
Select * from emp   where id = 12
        order by id
```

one reducer = key

① Row_number:
Row-number windowing function is used to get row number value with respective colomns ~~table~~.

Ex :

Table a

Name	
a	
b	
c	
d	d
e	e
f	

=>

Name ① Row

Name	Row no.
a	1
a	1
a	3
b	4
c	5
d	6
e	7

SK

Syntax:

```
Select name, Row_number over (order by
     Name) as Row
from table A ;
```

(ii) Rank:

Syntax:

```
Select name, Rank() over (order by
     name) as Row from table ;
```

Table A

Name
a
b
c
a
d
e
a
f

⇒

Dense Rank o/p

Name	Row
a	1
a	1
a	1
b	2
c	3
d	4
e	5
f	6

~~Dense~~ Rank o/p

Rank o/p

a	1
a	1
a	1
b	4
c	5
d	6
e	7
f	8

(iii) Dense Rank :—

Syntax :

```
Select name, Dense_Rank ( )
over (order by Name) as Row
from table A;
```

dense in o/p previous page table

INDEXS

Indexes is used to get "updated information of a particular "colomn" in a table.

Indexes are purely based on colomn oriented use cases.

Syntax :

```
Create Index CRCR ON
Table employ(salary) as
' org.apache.hivecompacthandler';
```

Syntax:

Create index crcp ON
Table emp(salary) as 'org.apache.hadoop.hive.*'

Purge: for deleting dat permanently from the trash (hadoop environment).

Syntax: purge Table –(Table name).

10.03.2017

SerDe

Serializier → object - byte

Deserializier → byte → object

SerDe is used to process unstructured and semistructured data in hive environment.

The usage of SerDe in hive is
- i) How to read data in hive &
- ii) How to write data into HDFS

Normally we have default SerDe's like
1) Reg~~ex~~ SerDe
2) Json SerDe
3) Avro SerDe

(Name, Salary, exp)

Syntax: to serde:

```
Create external table crcp
(name string,
   Salary int,
    exp int,
row format serDe 'org.apache.hadoop.
                hive.serDe2.Reg[illegible]rde'
with serDeproperties(" ____ ")
Location '/user/hive/crcp';
```

ACID:

A - Atomicity
C - Consistity
I - Isolation
D - Durrability

14 · March · 17

PIG.

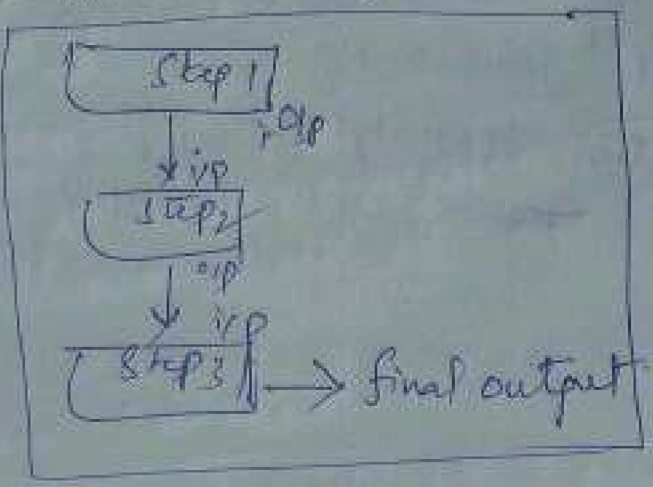
(OLAP)
↓MR
→hive

1> introduction to pig
2> pig architecture
3> Data types in pig
4> operaters and functions pig
5> parameter substitution pig
6> JOINS in pig
7> specilised joins in pig
8> optimization techniques in pig.
9> Types of ~~Gupta~~ UDF's in pig

pig is latin language it is used to process 3 v's of data i.e structured, unstructured & semi structured data, By using pig we can achive parallel processing..

pig latin language is program procedure language (PPL) (DAG)

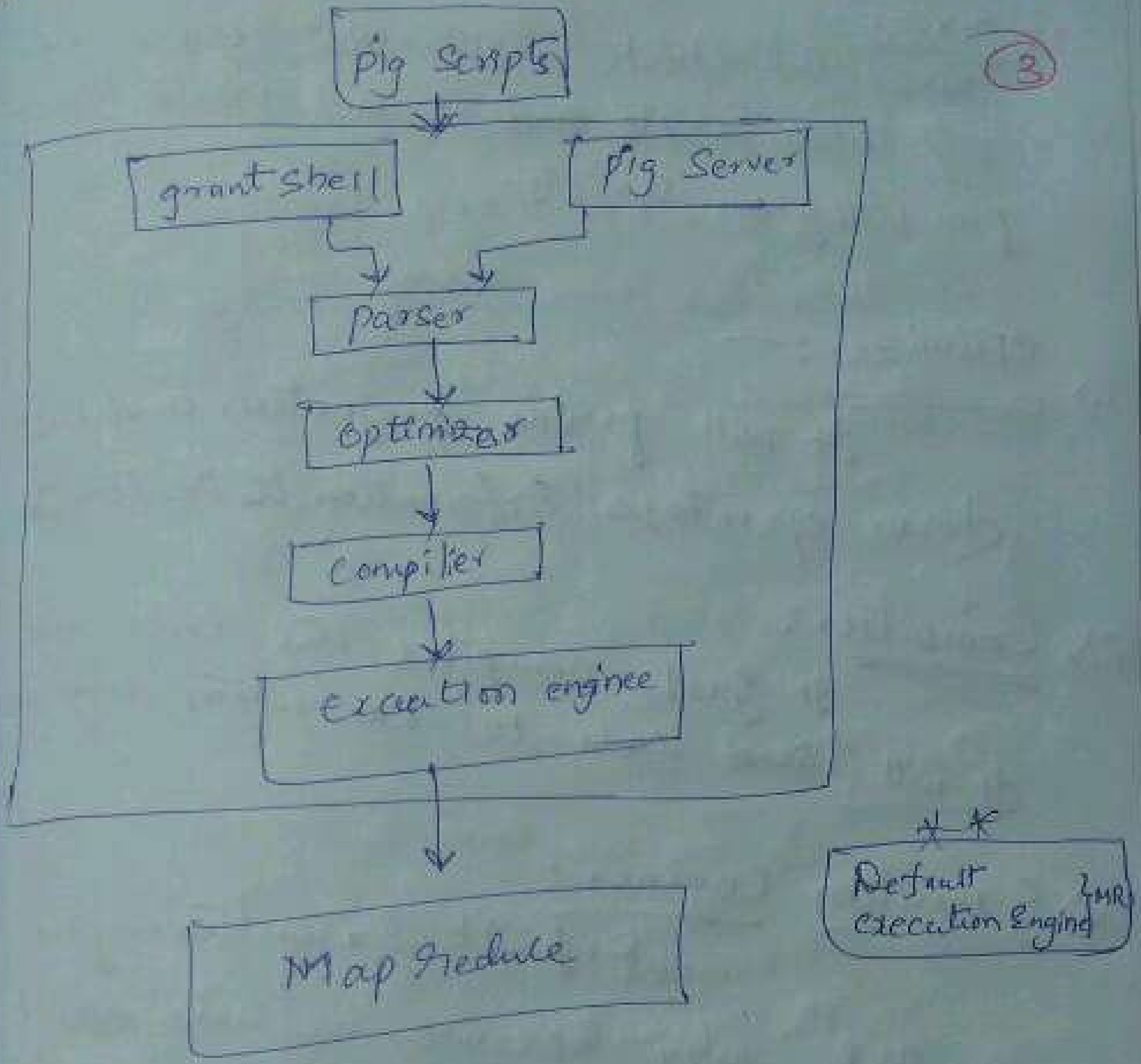
D - Direct
A - Acyclic
G - Graph



Architecture of pig:

If we want run pig script in real time on that to we have 2 options it is based on the vender and client requirement.

i) Grunt shell
2) pig server (connecting & through eclipse)



In pig Arctictecture we have 4 major steps

(POCE)

i>parser

ii) optimizer

3> compiler

4) Execution engines.

i) **parser:**
parser will check Syntaxtical errors of pig latin Scripts and it will create logical plan with given scripts.

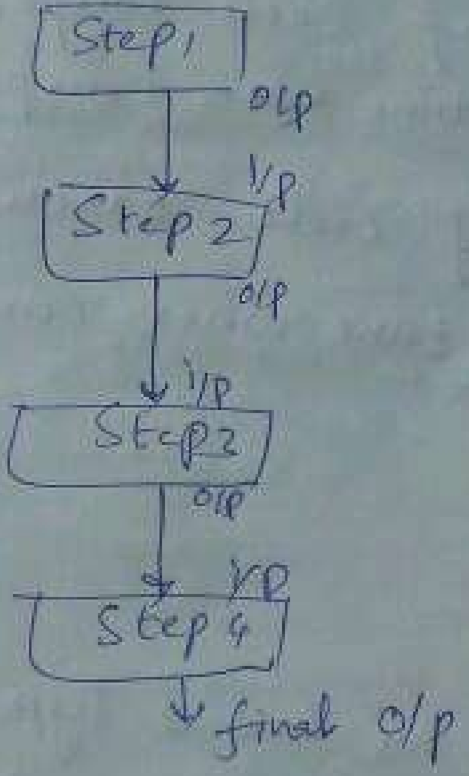
ii) **optimizer:-**
It will provide projection and push down operators information to the compiler.

iii) **Compilier:**
It gust compiles the code and it will send code to execution engine.

iv) **Execution engine:**
Default Execution engine is "MR" why because we have installed pig on top of hadoop on that to pig comes under processing (OLAP).

For processing in hadoop environment MR is master that's why pig will take default execution engine as a Mapreduce.

Direct Acyclic graph (DAG) / PPL



Q/n b/w hive / pig / MR interms of code composition and execution:-

→ In terms of code composition mapreducing will take more time when compared to hive and pig

→ In terms of code execution MR will take less time when compare to pig and hive.

If we compare pig & hive in term of code (composition/ execution) :-

⇒ In term of code composition pig will take less time when compared to hive

⇒ In term of code execution hive will take less time when compared to pig.

Data types :

In pig we have 2 types of data types.

i) Complex → Tuple, map, Bag

ii) premetive. → Int, double, chararay, bytearray, date type, boolean, float

i) Complex data types:

i) Field - ~~collection~~ piece of data ✓

ii) Tuple - 'collection of fields' - ('Hi', 'CRCP')

iii) Bag (or) Relations -

field ↗ Tuple

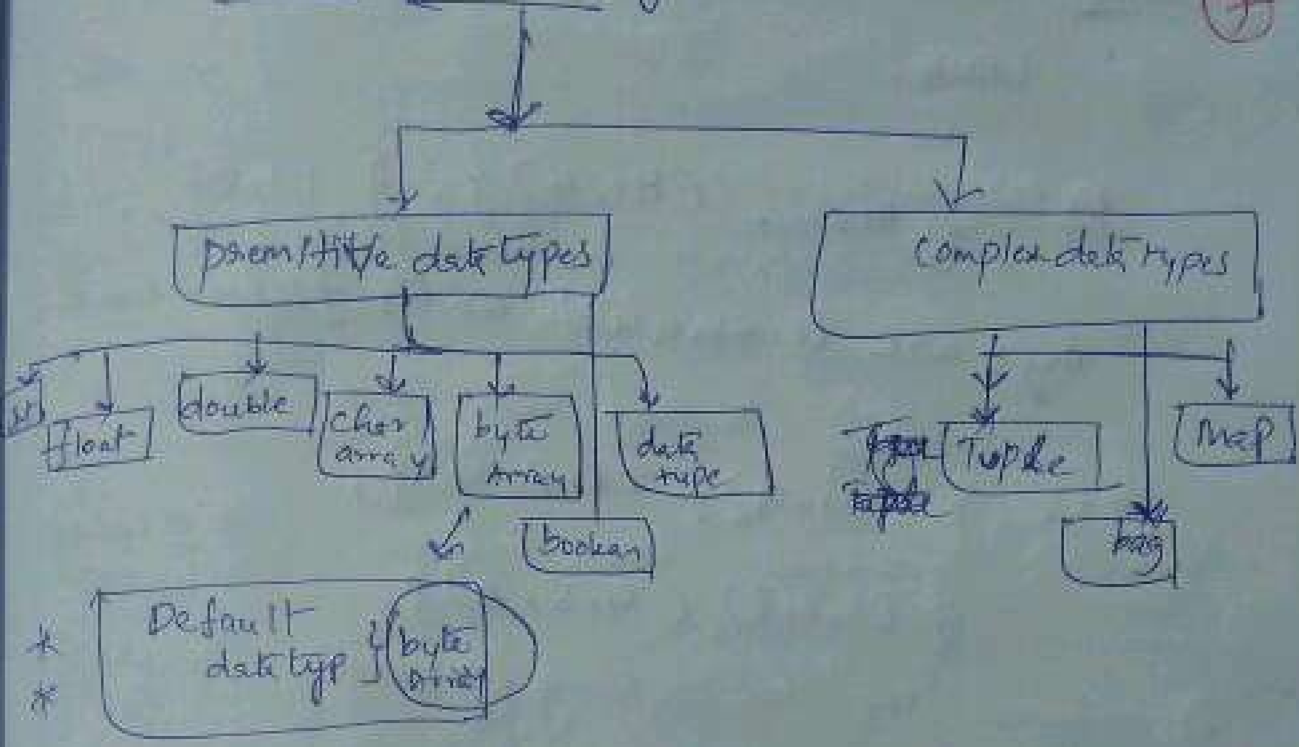
{('Hi' 'CRCP' 1,2,3,4) ('Hi', 'CRCP', them)}

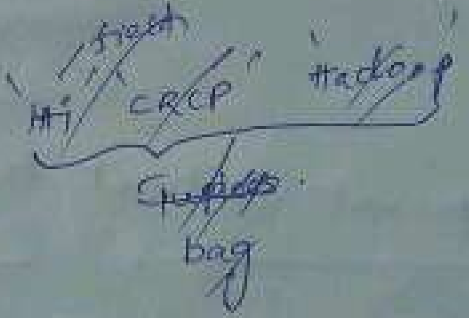
Bag.

map / combination of key value pairs [illegible] (Name # CRCP)

15.03.17

Data types in pig:





Field:

field is a piece of data Ex:- Hi, CRCP, Hadoop

'fields'

Tuple:

collection of fields is nothing but Tupple

(Hi, CRCP, Hadoop), (CHi, Hadoop)

Tuples

Note: Tuples are ends with brackets

3) Bag: collection of Tuples are nothing but bag.

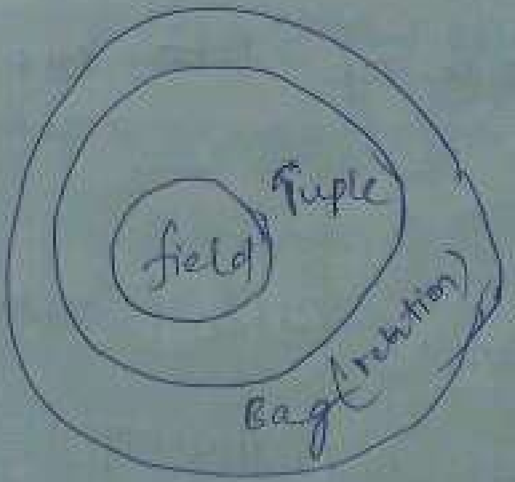
Ex: {(Hi,3), (Hi,4)}, {1,2,3}, (4,5,6)}

bags are terminated with flower brackets

Ex: {(Hi,4), (CRCP,4)},
{(1,2,3), (4,5,6)}

10 fields
4 Tuples

Superset subset relations


field
Tuple
Bag (relation)