

22/06/2015
Monday

right one

1 Name Node

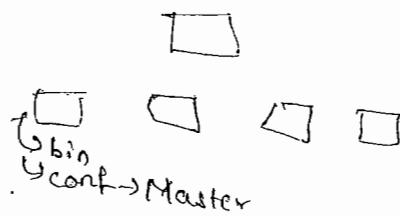
- (1) → In every mission we install ssh → Password less base protocol
- (2) → Generate the public key, and copied to the slave
- (3) → Putting ↘ RSA key

HADOOP 1.2.1

↳ bin → ENSURING 1.2.0
↳ config → All configuration files
↳ lib → dependency jar files

How to make master & slaves ?

- ④ Place Configuration →
this folder
- MASTER FILES
 - SLAVE FILES → 192.112.0.1
→ 192.112.0.2
→ 192.112.0.3
→ 192.112.0.4
- Master is a secondary node IP address.



Heartbit → the availability of data

Administration Tool → Every time admin don't going to login into each mission manually.

There come some tools, that is called administration tools.

- ① ↳ Apache Ambari → set up the cluster and monitor
- ② ↳ Cloudera Manager
- ③ ↳ Puppet
- ④ ↳ chef

Bin → bin/start.sh → sh → start mission

↳ address of the block in slave

We never start and stop the cluster in any time → in real time
where will we store

Header Node → slaves by IP address

Who will initiate data nodes → Master

- ⑤ - What is Heartbeat?

Heartbeat is used to wait for another IP address to come & join. For example - In a classroom, a teacher is waiting for student to all come and then start studying. In the same manner. Master is waiting to all the slave to join the cluster.

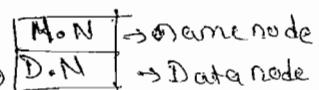
Slave → Master
Heartbeat

Fully Distributed Node

Each Main node is start with Daemon.

If each daemon running on separate JVM (fully distributed mode)

→ Give same IP address to slave and master.



sharing single JVM

→ Pseudo Distributed Mode → If all daemon sharing single JVM is called pseudo distributed mode

Master & slaves runs on same mission or JVM.

Hadoop-VM (Running) - Oracle VM VirtualBox

→ Bookmarks

↳ Installations

↳ Hadoop-1.2.1

↳ bin

↳ start-all.sh

go back

click on config

↳ Master → localhost → domain name

↳ Slave → localhost

/etc/hosts

127.0.0.1 → localhost

~\$ ls

~\$ pwd

~\$ cd

~\$ cd ..

~\$ cd ~ → root folder
↓ to /

~\$ cd ..

~\$ cd .. → present working directory

~\$ cat /etc/hosts

~\$ cd ~

~\$ pwd

home/vm4learning

~\$ ls

~\$ cd Installations/

~\$ ls

~\$ cd hadoop-1.2.1/

~\$ ls

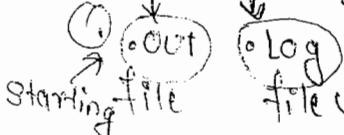
~\$ Start the cluster

bin

logs

?

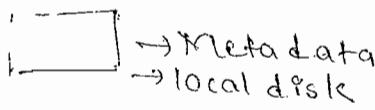
→ 2 types of file



How to format

data folder
↳ dfs

HDFS ?



~\$ rm -r logs/* → format your laptop
~\$ rm -r data/*
~\$ bin/hadoop namenode <
~\$ bin/hadoop namenode format
~\$

Now we start Hadoop cluster —

~\$ bin/start-all.sh
JPS → JVM processing system
~\$ JPS

~\$ bin/stop-all.sh

~\$ JPS → 4162 JPS
~\$ clear

~\$ bin/start-dfs.sh

~\$ JPS

~\$ bin/stop-dfs.sh

~\$ JPS

5233 JPS

~\$ clear

~\$ bin/start-mapreduce.sh

~\$ JPS

~\$ bin/stop-mapred.sh

~\$ bin/start-all.sh

~\$ JPS

localhost 50070 web confirm for name node

localhost 50075 — — for data node

goto web browser

localhost:50070

↳ Live Node - 01
↳ Dead Node - 00

PUT → Source to target

↓ Comment

↳ by using web console

20/6/2015
Monday

Home Work

What is Hadoop?

Apache™ Hadoop® is an open source software project that enables distributed processing of large data sets across clusters of commodity servers.

It is designed to scale up from a single server to thousands of machines, with very high degree of fault tolerance. Rather than relying on high-end hardware, the resiliency of these clusters from the software's ability to detect and handle failures at the application layer.

Hadoop Architecture -

Hadoop is composed of 4 components -

→ Hadoop Common

→ Hadoop Distributed File System(HDFS)

→ MapReduce

→ YARN (Yet Another Resource Negotiator)

Hadoop Common -

A module containing the utilities that support the other Hadoop components.

HDFS -

A file system that provides reliable data storage and access across all the nodes in a Hadoop cluster.

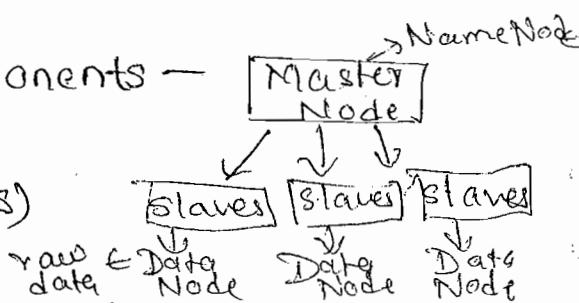
It links together the file systems on many local nodes to create a single file system.

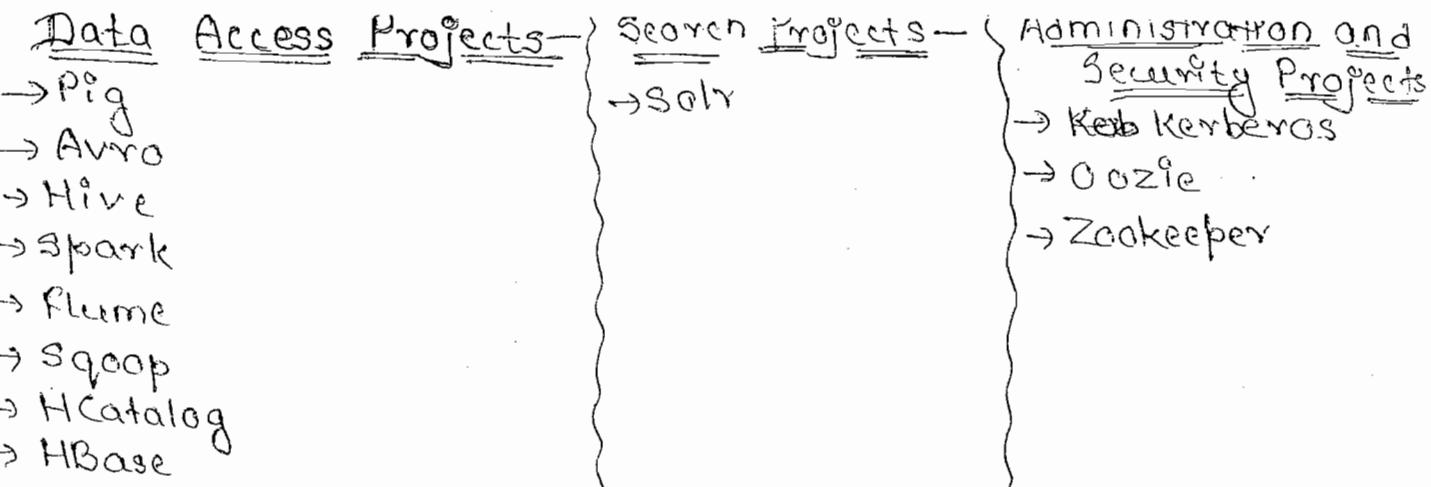
MapReduce -

A framework for writing applications that process large amounts of structured and unstructured data in parallel across a cluster of thousands of machines, in a reliable fault-tolerant manner.

Yet Another Resource Negotiator (YARN) -

The next-generation Map-Reduce, which assigns CPU, memory and storage to applications running on a Hadoop Cluster. It enables application frameworks other than MapReduce to run on Hadoop, opening up a wealth of possibilities.





What is SSH and HTTPS in the Hadoop Cluster

→ SSH and HTTPS can be used to transmit information securely

⦿ SSH (Secure Shell) is a secure shell that usually runs on top of SSL and has a built-in username/password authentication scheme that can be used for secure access to a remote host, it is a more secure alternative to rlogin and telnet.

⦿ HTTPS (HTTP Secure) is a HTTP running on top of SSL, adding security to standard HTTP communication.

SSH -

It is a good idea to use SSH for remote administration purposes (instead of rlogin, for example). But note that it is not used to secure communication among the elements in a Hadoop Cluster (Data-Node, NameNode, TaskTracker or YARN ResourceManager, JobTracker or YARN NodeManager, or the /etc/init.d scripts that start daemons locally).

The Hadoop Components use SSH in the following cases :-

- ⦿ The sshfence component of High Availability Hadoop configurations uses SSH; the shell fencing method does not require SSH.
- ⦿ Whirr uses SSH to enable secure communication with the Whirr cluster in the cloud.

HTTPS -

Some communication within Hadoop can be configured to use HTTPS. Implementing this requires generating valid certificates and configuring clients to use those certificates. The HTTPS functionality that can be configured in CDH4 is -

- Encrypted MapReduce shuffle (Both MRv1 and YARN).
- Encrypted Web UIs; the same configuration parameters that enable Encrypted MapReduce Shuffle implement Encrypted Web UIs.

RSA (cryptosystem)

Designers → Ron Rivest, Adi Shamir, and Leonard Adleman

First published → 1977

Certification → PKCS#1, ANSI X9.31, IEEE 1363

Key Size → 1,024 to 4,096 bit typical
Cipher detail

Rounds → 1

Best Public Cryptanalysis.

RSA involves a public key and a private key.

What is Hadoop?

→ Hadoop is analysis for Bigdata.

→ Hadoop is an open source framework for creating distributed applications that process huge amount of data.

One definition of Huge -

Ex → 25,000 machines

more than 10 clusters

3 PB of data (compressed, com-replicated)

700+ users

10,000+ jobs/weeks

other definition

→ Hadoop is a open-source, distributed, batch-processing and fault-tolerance system which is capable of storing huge amount of data [TB, PB, zettabytes - etc] along with processing data on the same amount of data.

→ Hadoop easy to use parallel programming model.

→ Hadoop framework consists two main base components.

① HDFS

it is responsible for storing massive amount of data on the cluster.

② MapReduce

↓
It is responsible for processing massive amount of data on the cluster.

⇒ HDFS & MapReduce capabilities are it's kernel for Hadoop.

23/06/2015

Tuesday How to place a file in Hadoop cluster?

\$ ls Where is Hadoop? in Installation.
\$ cat .bashrc

- There is .bashrc file in there
which is Hidden file

if you want to direct access

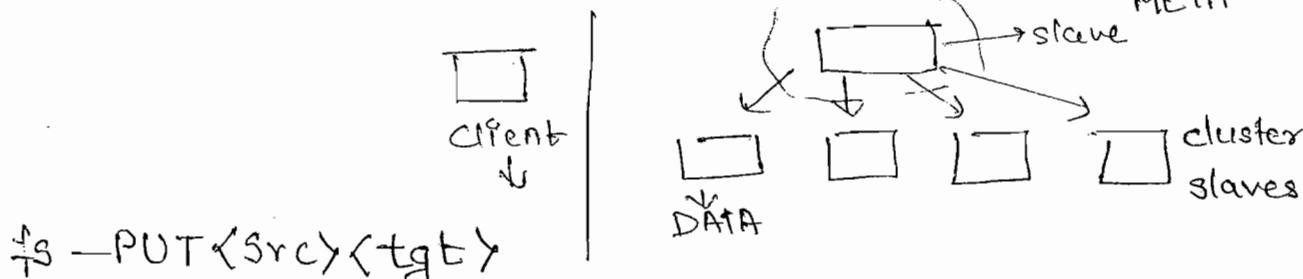
\$ cd \$ HADOOP_HOME
\$ pwd

Start the cluster →
\$ jps

Runs a Genuien file system with Hadoop?

\$ bin/Hadoop fs

Now we talking about placing a file ^{is} nothing but put command.
config files → 84 MB



fs -PUT <Src> <tgt>

Three Types of Configuration files

- ① HDFS-SITE.XML → ^{search in google} replication → dfs.replication - 3
- ② CORE-SITE.XML
- ③ MAPRED-SITE.XML

dfs.replication - 3
max size → 128 MB
min size → 64 MB

block size → 64 bytes

- These are called system level configuration files.
- which comes with Hadoop in the config folder.
- Every change value of file will be done by the property in config.
- If you want to overwrite any of the property, we have to change with system level configuration.
- If we want to change block size follow same rule.

What is the default folder in Hadoop Cluster?

/user/Vmulearning → Hadoop

/user/System
/home/dr → local

If we want to divide client into b₁ and b₂

→ dfs.name.node.name.dir →
Determines where on the local filesystem the dfs.co

`hadoop.tmp.dir`

↳ where we do all work

data → folder Here the actual meta-data store...

↳ `dfs`

↳ name → in these location metadata get stored

`dfs.datanode.data.dir` → Determines where on the local system file
an DFS data node should store its blocks.

Go to Workspace

↳ big data

↳ HBase-Bulkimport

↳ input

`fs -PUT location Rowfeeder.csv`

(user/vm4learning)

the
three config file

why we use comments → to take or get environment.

`$ bin/hadoop fs -help put` → what is meant by put?
copies files from the local file system
into fs.

right click on `Rowfeeder.csv`

↳ properties

↳ location

↳ `bin/hadoop fs -put /home...`

location is copied. ↪

⇒ By these we divided the file into 2 blocks.

file

blocks

Home | Installation | hadoop-2.0.1 | data | dfs | data | current → You can
see 'divided block' in these...

Go to → Hadoop NameNode localhost...

`http://localhost:50070/dfs/health.jsp`

Creating Directory in Hadoop

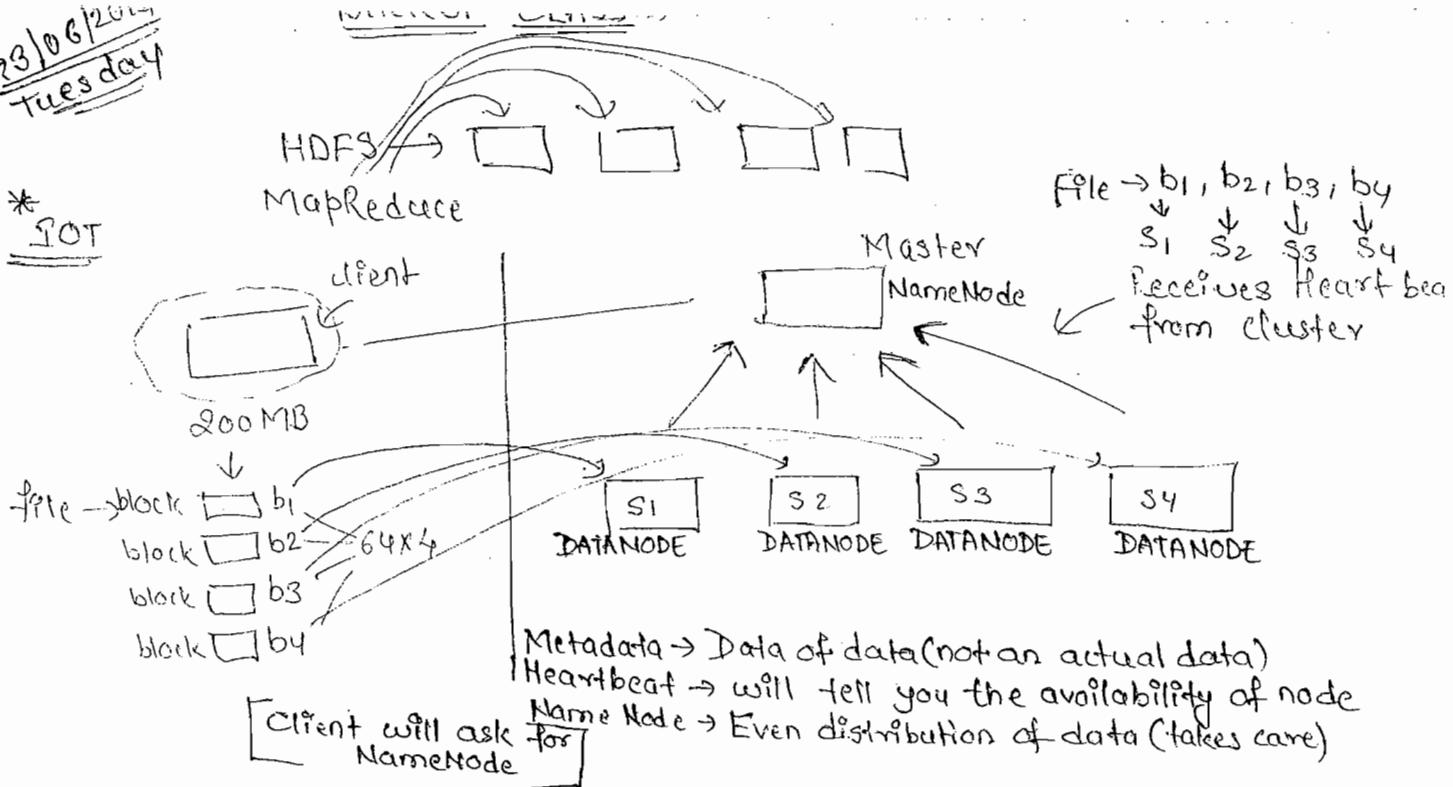
`$ bin/hadoop fs -mkdir /user/vm4learning/Naresh` ↪

New dir called Naresh been created

HDFS: /user/vm4learning/localhost

23/06/2014
Tuesday

* IOT



\$ bin/hadoop fs -mkdir huma

HDFS (Hadoop Distributed File System)

Hadoop modes of Installation -

- ① Standalone mode
- ② Pseudo Distributed mode [Labwork]
- ③ Fully Distributed mode [Realtime]

1) Standalone mode -

→ Development

→ Test

→ Debug

- Single machine
- No Daemons are running.
- Everything runs in single Jvm
- Standard OS storage.

→ Good for development and ~~test~~ with small data, but will not catch any errors.

2) Pseudo Distributed mode - [multiple jvms in single mode]

→ Single machine but cluster is simulated.

→ Daemons run in separate jvm.

→ Separate jvm's in single node.

→ Good for Development and Debugging.

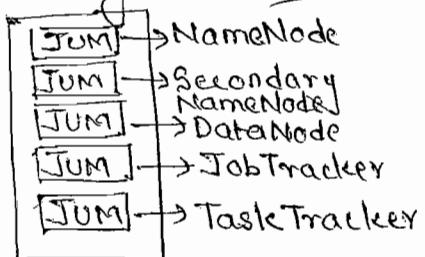
3) Fully Distributed mode - [All components run in... Separate Nodes]

→ Run Hadoop on cluster of machines

→ Daemons run

→ Production environment

→ Good for staging & production.



What is Distributed File System:-

→ System that permanently store Data.

→ Support concurrency, distribution, replication access to file and remote servers.

→ Divided into logical units(files, shards, chunks, blocks).

→ DFS's are working network based approach because of DFS's are more complex than regular disk file systems.

Example - File system tolerate node failure without suffering Data loss.

Hadoop Distributed File System:-

→ Hadoop is a distributed file system and it uses to store bulk amounts of data like terabytes can even petabytes.

→ HDFS support high throughput mechanism for accessing this large amount of information.

→ In HDFS, files are stored in sequential redundant manner over the multiple machines and this guaranteed the following ones -

(1) Durability to failure.

(2) High availability to every parallel applications.

- Example: NFS (Network File System)

→ NFS gives remote access to a single logical volume stored on a single machine.

→ NFS server can visible a portion of it's local files system to external clients and also the client can mount this remote file system directly into their own linux file system, and interact with it as though it were part of the local drive.

Advantage of NFS -

→ It is transparency [That is clients do not need to be particular aware that they are working on files stored remotelut -

ADVANTAGES OF HDFS -

(1) HDFS store large amount of information.

(2) HDFS is simple and robust coherency model.

(3) That is it should store reliability.

(4) HDFS is scalable and fast access to this information and it also possible to serve large number of clients by simply adding more machines to the clusters.

(5) HDFS should integrate well with Hadoop mapReduce, allowing data to be read and computed upon locally when possible.

(6) HDFS providing streaming read performance.

(7) Data will be written to the HDFS - once and then read several times.

(8) The overhead of cashing is helps the data should simply be re-read from HDFS source.

(9) Fault-tolerance by detecting faults and applying quick, automatic recovery.

(10) Processing logic close to the data, rather than the data close to the processing logic.

(11) Portability across heterogeneous commodity hardware and OS.

(12) Economy by distributing data and processing across clusters of commodity personal computers.

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

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

Disadvantages of HDFS -

- 1) In distributed file system, it is limited in its power.
- 2) The files in an NFS volume all reside on a single machine. This will create some problems.
- 3) It does not give any reliability guarantees if that machine goes down.

Example - By replacing the files to other machine.

- 1) All the clients must go to this machine to retrieve their data. This can overload the server if a large no. of clients must be handled.
- 2) Clients need to copy the data to their local machines before they can operate on it.

Goals of HDFS -

- 1) Very large distributed file system -
10K nodes, 100 million files, 10PB.
- 2) Assume Commodity Hardware -
Files are replicated to handle hardware failure. Detect failures and recover from them.
- 3) Optimized for Batch processing -
Data locations exposed so that computations can move to where data resides. It provides very high aggregate bandwidth.

HDFS is a block Structured File System -

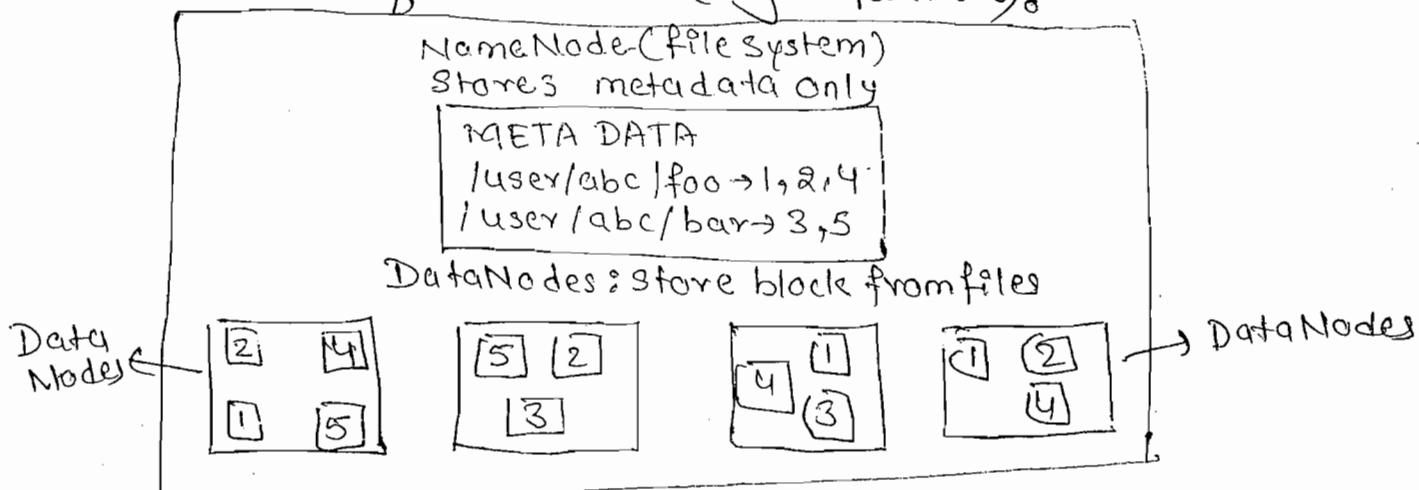
Block -

Block is a minimum unit of data that said in HDFS, which is typically 64mb by default.

However we can increase the block size in the multiples of 64mb.

- Each file is broken into block of a fixed size of 64mb. These blocks are small across a cluster of one or more machines with data storage capacity.
- Individual machines in the cluster are called the data nodes.
- A file can be made of several blocks and not necessarily stored on the same machine.
- The target machine choose each block randomly on a block-by-block basis.

- Q6, access permission to a file may read the co-operation of multiple machines & it supports file size for larger than a single machine DFS.
 - Individual files sometimes need large space than a single hard drive could hold. If several machines must be involved in the serving of a file then a file could be rendered unavailable by the loss of any one of those machines.
- HDFS combats this problem by replicating each block across a no. of machines (by default 3).



Features of HDFS :-

- HDFS is a file system designed for storing. Have some key characteristics. They are-
- ① Support for very large files.
 - ② Commodity Hardware
 - ③ Streaming data access
 - ④ High-latency data access
 - ⑤ Lots of small files
 - ⑥ Multiple writers, arbitrary file modifications.
 - ⑦ Moving computation is than moving data.

10/6/2015
Wednesday

\$ bin/start-all.sh

\$ rm -r logs /* ↳ format logs folder

\$ rm -r data/* ↳ format data folder

\$ bin/hadoop namenode-formatted ↳

Stopping the cluster -

\$ bin/stop-all.sh

Now start it

\$ bin/start-all.sh

\$ bin/hadoop fs -mkdir /user/vm4/learning/Namesh

Go to name node → Browser

\$ bin/hadoop fs -help mkdir

\$ bin/hadoop fs -mkdir Satya

Go to NameNode → Browser

→ metadata is first updated in edit logs

~~put~~ \$ bin/hadoop fs -put ^{space} /RowFeeder.csv ^{location} /user/vm4/lea^{nning/Rowfeeder.csv}

See → Home ~~as current~~

\$ bin/hadoop fs

\$ bin/hadoop fs -help ls → what is meant by ls?

\$ bin/hadoop -ls

\$ bin/hadoop fs -ls /user/vm4/learning/ ↳ 3 files

↳ what is in my cluster
it will show

\$ bin/hadoop fs -ls

\$ bin/hadoop fs -help ls

\$ bin/hadoop fs ls ↳

\$ bin/hadoop fs -ls /user/vm4/learning/ ↳ 3 files

\$ bin/hadoop fs -help du ↳

\$ bin/hadoop fs -du ↳ disk usage
file tabel

\$ bin/hadoop fs -du /user/vm4/learning/Namesh/Rowfeeder.csv ↳
gives actual path of file

\$ bin/hadoop fs

\$ bin/hadoop fs -help du ↳

\$ bin/hadoop fs -du

du → for directory

\$ bin/hadoop fs

\$ bin/hadoop fs -help count ↳

\$ bin/hadoop fs -count ^{4th} ↳

\$ bin/hadoop fs -count /user/vm4/learning/Namesh/Rowfeeder.csv ↳

count → no. of files in
specific folder/path

10L

Count

\$ bin/hadoop fs

\$ bin/hadoop fs -help -help mv[src]>[dst]

\$ bin/hadoop fs -ls /user/vm4learning {cut & paste}

\$ bin/hadoop fs -mv

applies only
4 wives where
1 husband when
what
How


mv is used to move
file to another
within cluster
Do not use in client

\$ bin/hadoop fs -mv /user/vm4learning/Naresh/

RowFeeder.csv /user/vm4learning/
/satya/ RowFeeder.csv

To

columnfeeder.csv

Change ⇒ Metadata changes or gets updated...

\$ bin/hadoop fs

copy paste

\$ bin/hadoop fs -help cp

columnfeeder.csv
n

\$ bin/hadoop fs -cp /user/vm4Learning/satya/RowFeeder.csv
/user/vm4Learning/satya/RowFeeder.csv

\$ bin/hadoop fs -rm [-skipTrash] <path>

\$ bin/hadoop fs -help rm datanode is stored in local disk

\$ bin/hadoop fs -rm /user/vm4learning.csv/RowFeeder.csv

\$ bin/hadoop fs -rm -help rmy

\$ bin/hadoop fs -rmy /user/vm4learning/Naresh/
RowFeeder.csv

rmy

\$ bin/hadoop fs -rmy /user/vm4learning/satya/
columnfeeder.csv

\$ bin/hadoop fs -count /user/vm4learning/

\$ bin/hadoop fs -ls } if give same file

\$ bin/hadoop fs -lsr } then there is no files

\$ bin/hadoop fs -help copyfromlocal

copy from <localsrc> --- <dst>

\$ bin/hadoop fs

{ we don't use in real time } \$ bin/hadoop fs -help -movefromLocal <localsrc>...<dst>

↓
delete except that the src is deleted after it's copied.

\$ cd \$HADOOP_HOME

~~10/6/2014~~
~~Wednesday~~
~~Thursday~~

Write → put
Read → get

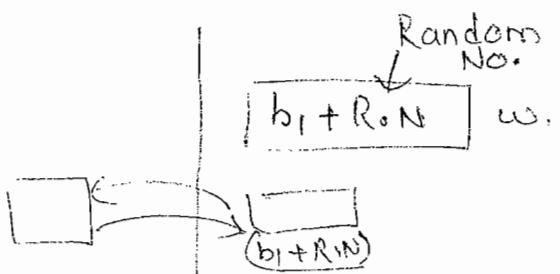
→ Hadoop is Batch Based not Web based.
→ Web based access is not present because processing of big big data is not possible for so many online users parallelly.

Why you need to get a file?
Where it was?

\$ bin/hadoop fs -put [location] /Rowfeeder.csv /user/vmfilelearning/Mareesh/Rowfeeder.csv

PUT <src> <target>
client cluster

GET <src> <tgt>
cluster client



- Hadoop is a ~~cluster~~ filesystem
we can write file at once and read it many time.
→ We can only read, we can append but we can't update, insert or delete.

Command —

-get[ignorecrc][Ecrc]<src><localdst>]

\$ ls
\$ pwd
\$ bin/hadoop fs -help get

```
$bin/hadoop fs -get /user/vm4learning/Naresh/Rowfeeder.cs
```

```
$pwd
```

```
$bin/hadoop fs -get /user/vm4learning/Naresh/Rowfeeder.csv /home/vm4learning/Desktop/HDFSGET/Rowfeeder.csv
```

On desktop

Create New folder - HDFSGET

```
$bin/hadoop fs -put /home/vm4learning/workspace/bigdata/HBase-BulkImport/input/Rowfeeder.csv /user/vm4learning/Naresh/ColumnFeeder.csv
```

↳ Desktop

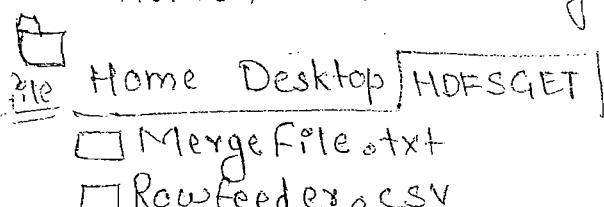
↳ HDFSGET → property

getmerge —

```
$bin/hadoop fs -help getmerge
```

-getmerge <src> <localdst> → Get all the files in the directories that matches...

```
$bin/hadoop fs -getmerge /user/vm4Learning/Naresh/ /home/vm4Learning/Desktop/HDFSGET/Mergefile.txt
```



cat → To concatenate files within the cluster

```
$bin/hadoop fs -cat /user/vm4learning/Naresh/Rowfeeder
```

-text <src>

\$bin/hadoop -help text → converts Binary data into text(human readable format) we can read zip file also & TextRecordInputStream

CopyToLocal

\$ bin/hadoop fs -help copyToLocal

→ Identical to the -get command

↳ How data will be retrieve with the help of robots from the cluster.

-setrep [-R] [-w] <rep> <path/file>

General Commands

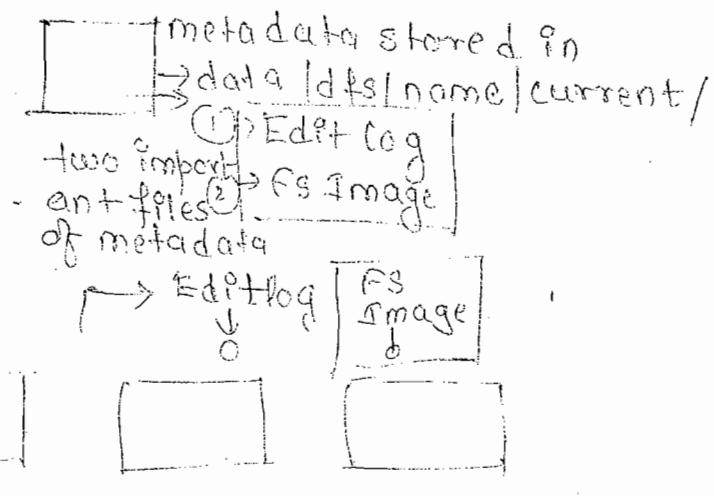
[-touchz <path>]

[-test -[ezd] <path>]

[-stat]

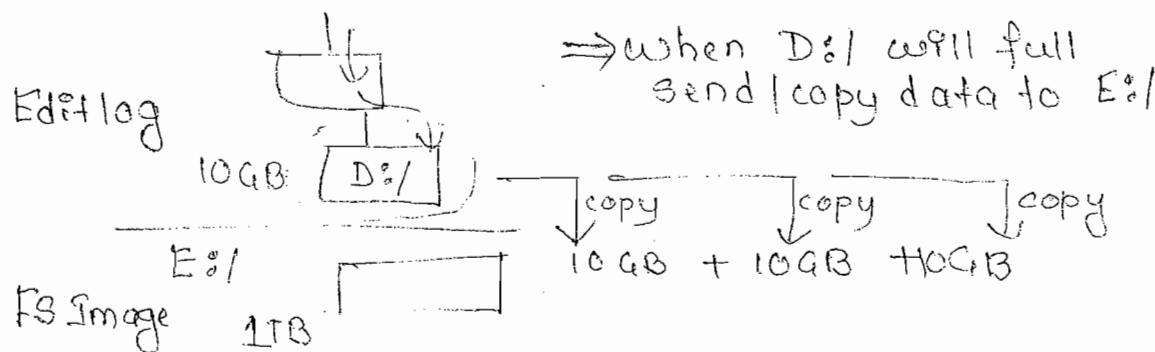
[-tail]

Client



Home Installation Hadoop 2.1 data dfs name current
Create a directory

- When a client update or added some data the log edit log get updated.
- The Every activity 1st record in edit log
- F.S.Image is actual Metadata.



→ As the fs image is usage huge file.

→ 1st edit in edit log

→ When edit log full
then it is going to the fs image.

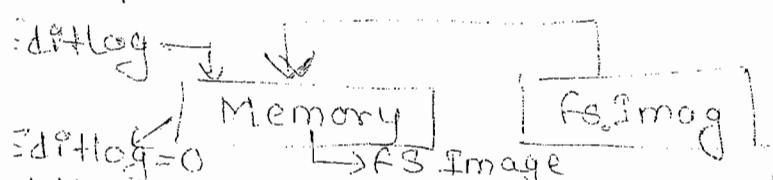
→ 1st → Stop the cluster

→ 2nd → Start the cluster

Now we see by example

① Start - all .sh

whenever you start cluster, the actual fs image goes to the memory



→ Whenever you restart the editlog also goes in / loaded into the memory.

→ When editlog also copied, the edit log becomes free
Editlog = 0

→ When we stop the Hadoop cluster,
the ~~RS~~ Memory data goes to the FS Image.

This is drawback too

→ The data is loading into the Memory & it takes lots of time (2 days)

→ Why we have to do this?
⇒ To free edit log

When edit log is copied till it is copied no other operation will run.

→ Every time we have to stop & start the cluster.

Secondary NameNode is a Housekeeping Node for our Namenode which increases the performance of the namenode

→ help to Secondary NN job is to merge the editlog ^{& fs image} at regular interval w/o stopping cluster.

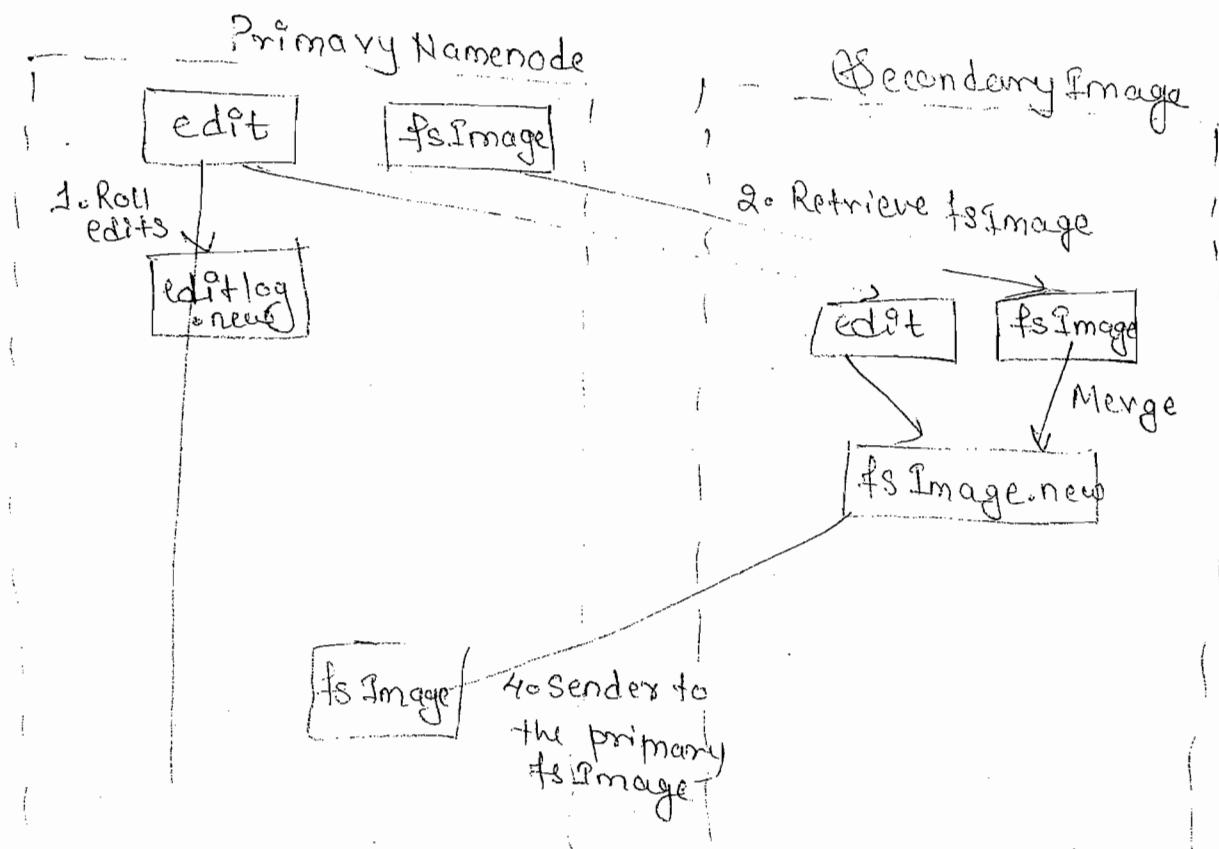
- ⇒ Secondary NN job is to free editlog.
- ⇒ It is not an actual backup log, when your nn editlog is 0, then the namenode lost then only it is known as backup node.

⇒ But it is never called as backup node.

⇒ Secondary NN checks the size & time of editlogs.
everytime it check

① Size | 10 GB

② Time



Secondary nn is a checkpoint node to the hadoop cluster
⇒ It is not at all a backup node.

* Let see the web console of secondary namenode
whether it is working or not...

[localhost:50090] ⇒ for Secondary NameNode

3600 sec → 60 min

\$ sudo init 0 → stop

(super user do)

\$ sudo init 6 → start

6/6/2015
Friday

REPORTS

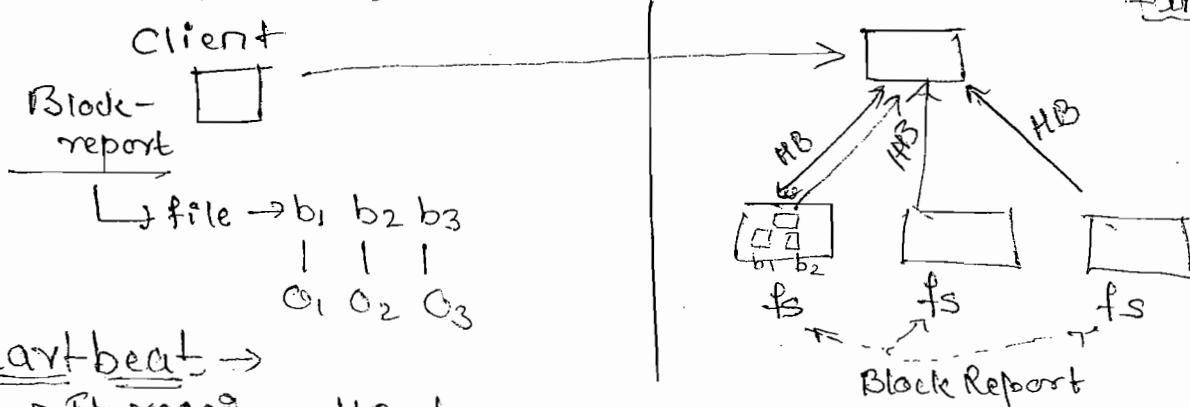
① Heartbeat

② Block Scanner Report

③ Block Report

} Three types of reports -

Common block diagram for 3 types of report showing functionality by each report



Heartbeat →

→ It receives HB from all slaves.

→ HB gives the availability of datanode.

→ Check live nodes.

Data Block Scanner Report

- Data node gives the health report (status) to the namenode called Data block scanner Report.
- Block level Data block scanner report is in
- What the data node contains, whether it is important or not, this come under health.

`http://localhost:50075/blockScannerReport`

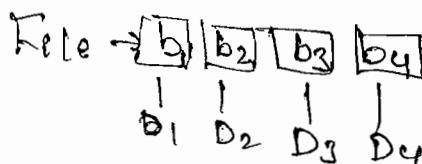
`http://localhost:50075/blockScannerReport?listblocks`

→ It gives more detail of data block scanner report

Home Install Hadoop data -- - - - - - Current

Block Report

- Block Report is in file level.

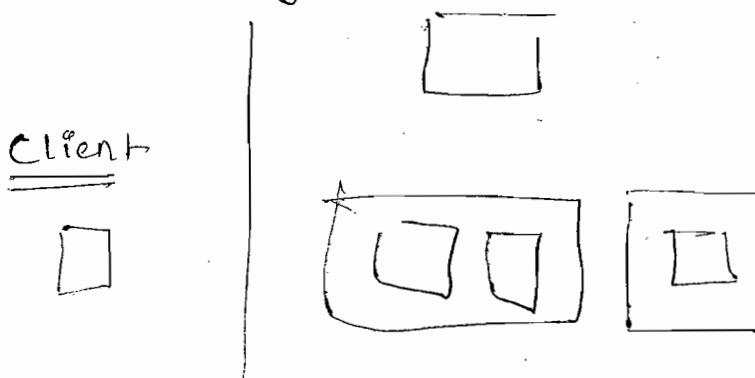


- Construct with the MB and Data block scanner report.
- It gives the health of the file.

→ File system checkability (fsck) →

`$ bin/hadoop fsck`

- which going to check block report.



bin/hadoop fsck -blocks -locations -racks filename

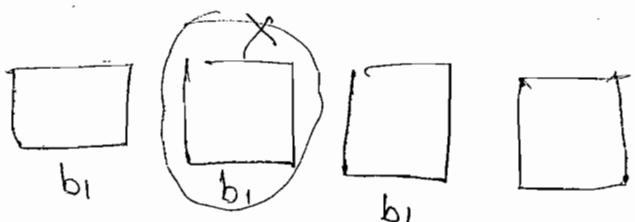
→ racks → how many datanode has been distributed

bin/hadoop fsck -blocks -locations -racks /user/vm4learning
/Naresh/Rowfeeder.csv

It gives the status: HEALTHY

No. of data nodes: 1

↳ comes from metadata



Actual replication is 3

Assume that 1 data block is lost then -

it shows → 2

therefore it is Under-replicated block → is a problem

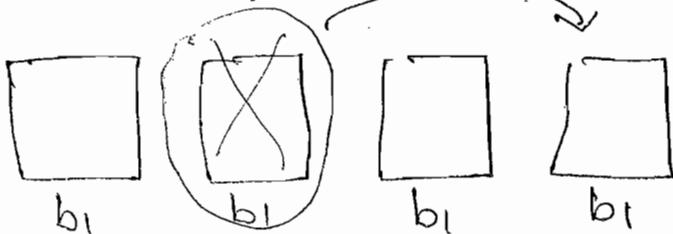
Balancer in Hadoop cluster

↳ Under-replicated block → 2

↓ convert it to

Original-replicated block - 3

By use of balancer

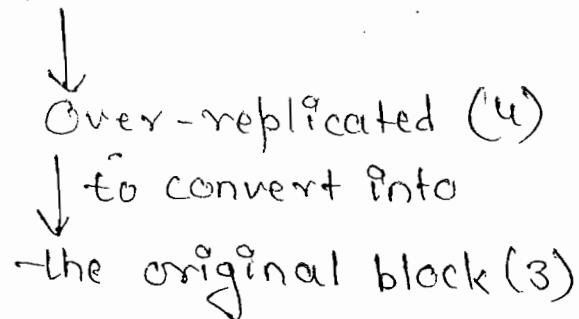


Balancer

⇒ It takes care, the data node should be evenly distributed.

Right now \Rightarrow it is over-replicated block
this is also a problem

now, again we use Balancer



Missing-Replica -

How many blocks are missing from the actual (Original block) in under-replicated, is known as Missing-Replica.

original replication
 1 > 10

9 is missing-replica.

\rightarrow bin/hadoop fs

-D ~~ed~~ \Rightarrow will overwrite any of config.

\$ls

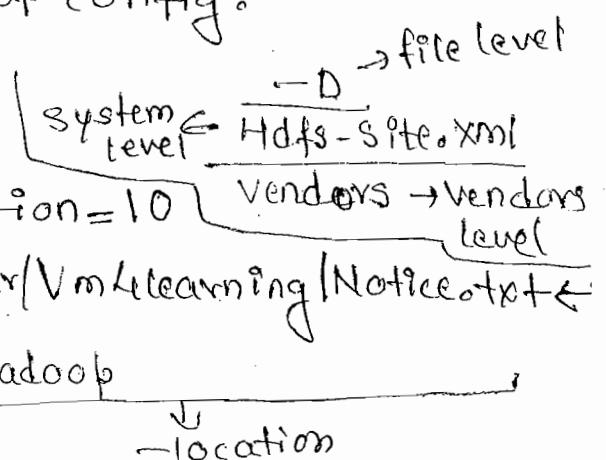
NOTICE.txt

\$bin/hadoop fs -D 'dfs.replication=10'

~~not~~ hadoop input NOTICE.txt | user/Vm4learning/Notice.txt

\$bin/hadoop fsck -blocks

-racks filename.



bin/hadoop fsck -blocks -locations -racks /user/vm4learning
/NOTICE.txt ↪

bdfs-site.xml

Srinu.xml

dfs.rep

10

dfs.namenode.name.dir →

/home/vm4learning/desktop/NN → c

dfs.namenode.data.dir →

/home/user/desktop/DN →

\$ bin/hadoop fs -conf /home/vm4learning/desktop/
configuration -put

\$ bin/hadoop fs -Ddfs.replication=10 -put /home/vm4learning
/Satya - - - - -

Setrep ⇒

How to change the existing file properties
or overwrite the replication

\$ bin/hadoop fs -help setrep

\$ bin/hadoop fs -setrep 20 /user/vm4learning/Namesh/
Rowfeeder.csv

\$ bin/hadoop fsck
-help

\$ bin/hadoop -fsck -move {destination}

⇒ It moves all the corrupted data to the destination.

~~39/06/2016~~

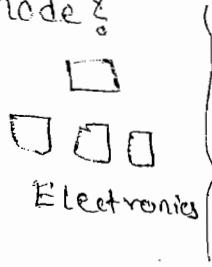
~~Monday~~

Why we have to add datanode?

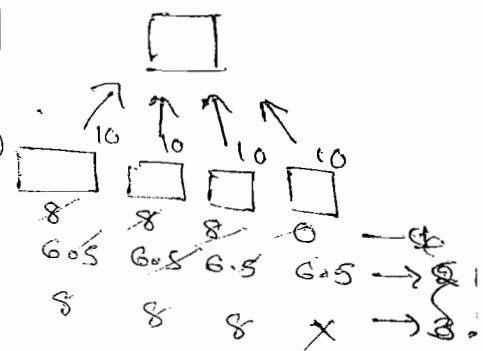
① How to add new datanode?

with
stopping
cluster

① Stop



Fashion



without
stopping
cluster

① Sminux.txt

↳ IP → 192.168.0.4

② Hdfs-Site.xml → dfs.hosts →

a list of hosts that are permitted to connect to the namenode.

{property}

{name}dfs.html{name}

{value}SIRNU.txt{/value}

↑
full location

{property}

③ Refresh to namenode

④ start-balancer.sh

② How to decommission in datanode?

with
stopping
condition

How to decommission in datanode?

① stop-all.sh

1
2
3
*

{name}property

{name}dfs.host.exclude{name}

{value}Regnuma.txt{/value}

{property}

with
stopping
condition

① Huma.txt

↳ 192.168.0.4

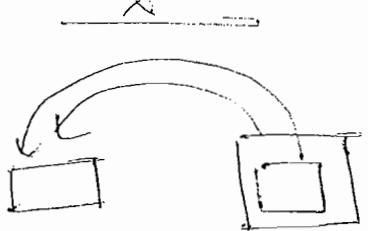
② Hdfs-site

dfs.exclude

Name a file that contains a list of hosts that are not permitted to connect to the name node.

③ Refresh N.N

④ Start-balancer.sh



Offline Xwrm
Read update

→ Whenever you start your cluster the time taken to load fs image into mem will be high, in this time the system will be offline, where we can't update to the metadata but we can read the data.

→ The mode we can't perform any updates to the metadata is called safemode

→ Usually we will place our cluster into safemode when we go for maintenance.

\$ bin/hadoop -safemode get

\$ bin/hadoop dfsadmin -safemode get

\$ bin/hadoop dfsadmin -safemode enter ✓

\$ bin/hadoop dfsadmin -safemode enter ✓

⇒ In these, we can not perform any updates & writes

\$ bin/hadoop fs -put CHANGES.txt /user/vm4learning/
CHANGES.txt ⇒ error

came out of safemode

\$ bin/hadoop dfsadmin -safemode leave //

Safemode is OFF

\$ bin/hadoop fs -put CHANGES.txt /user/vm4learning/
CHANGES.txt

\$ bin/hadoop fs -get /user/vm4learning/CHANGES.txt
\$ mv - safemode.txt

~~30/06/2014~~ --- going to talk about all dfsadmin operations -
~~tuesday~~ \$ bin/hadoop dfsadmin

① Report -

\$ bin/hadoop dfsadmin -help report

report will be give basic filesystem statistics.

\$ bin/hadoop dfsadmin -report

② safemode - \$ bin/hadoop dfsadmin -safemode enter

③ SaveNamespace -

\$ bin/hadoop dfsadmin help saveNamespace

| 1st → turn your clusters into safemode

| 2^d → then apply →

\$ bin/hadoop dfsadmin -saveNamespace

Now come out of the safemode

④ RefreshNodes -

dfs.hosts, dfs.hosts.exclude
↓
(refresh both)

\$ bin/hadoop dfsadmin -refreshNodes

⑤ finalizeUpgrade -

versionfile -

\data\dfs\name\current\

\$ bin/hadoop namespaces

VERSION.txt

upgrade

→ previous & current status

\$ bin/hadoop namenode -upgrade

Previous
Current

\$ bin/hadoop dfsadmin -upgrade

progress status ←

for
force

If we have issue, we will rollback it →

```
$ bin/hadoop namenode -rollback
```

If you want to commit:

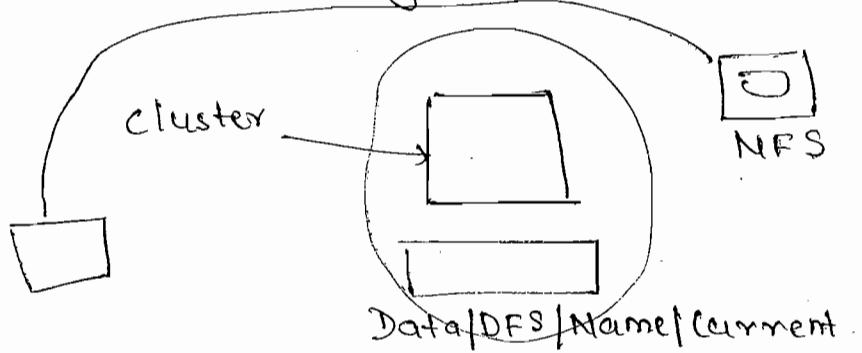


in hdfs → we have command called
"finalizeUpgrade"

```
$ bin/hadoop dfsadmin -finalizeUpgrade
```

```
$ bin/hadoop dfsadmin -importCheckpoint
```

How will you have the exact location in the cluster?



The time taken to load the fs image is very big problem.

`<property>dfs.namenode.name.dfs`
`<value> NFS system`

There is a framework called zookeeper which can handle the situation.

⑥ metasave —

```
$ bin/hadoop dfsadmin -help metasave ↴
```

\$ bin/hadoop dfsadmin -metasave srujan.txt

Goto \Rightarrow

↳ Installation

↳ hadoop 1.2.0

↳ logs

↳ srujan.txt

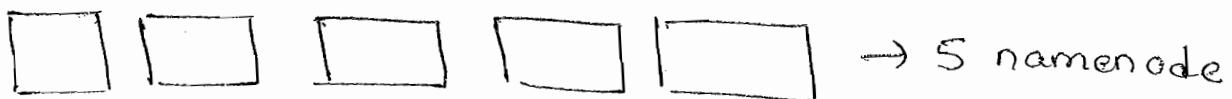
(7) -setQuota <quota> <dirname> - - -

\$ bin/hadoop dfsadmin -help setQuota

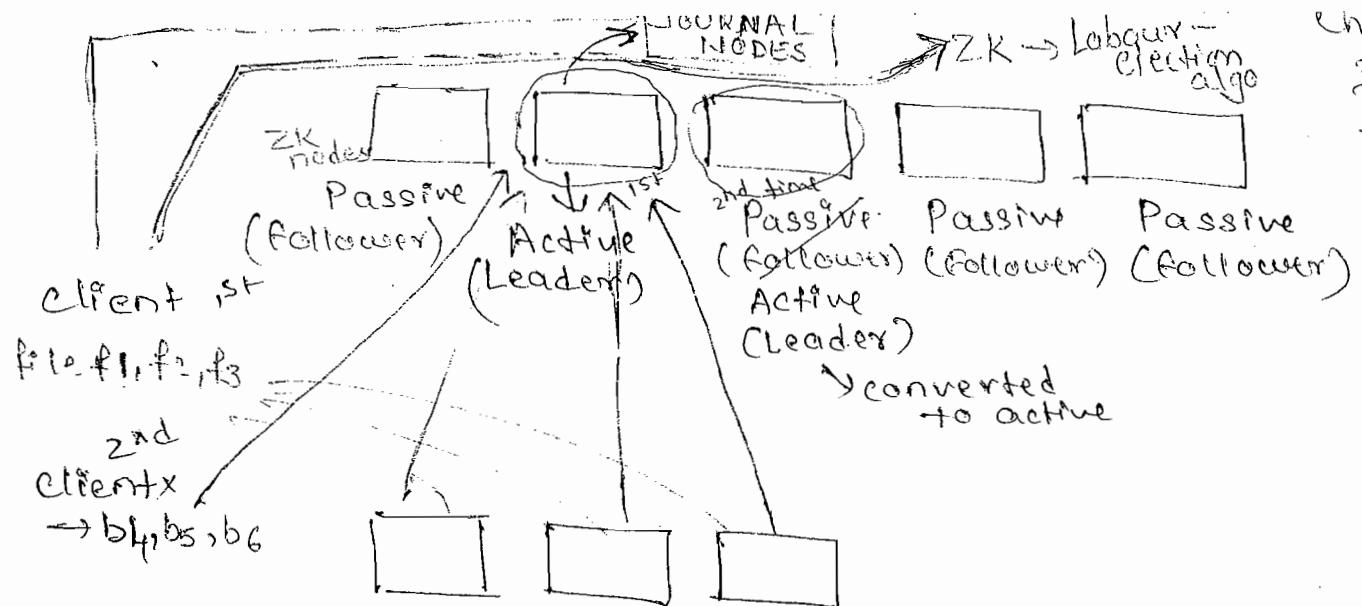
===== X =====

Zookeeper

- is a tool which is used to provide high availability of data namenodes
- is a distributed framework
- is a coordinating, distributed framework.
- Zookeeper provides high availability in the context of hdfs.
- Zookeeper is also called as a data discovery tool.
- Zookeep contains odd number of namenodes



- Zookeep will run the ^{algorithm} ~~will~~ a called leader-election algorithm.
- It select randomly a name node as an active node.



→ Client will always contact to zookeeper for active namenode

→ How contains active namenode → zookeeper

→ Second time when the client again sends the blocks, then it directly contact to the Active node.

→ In 1st time only, the client contacts directly to the zookeeper.

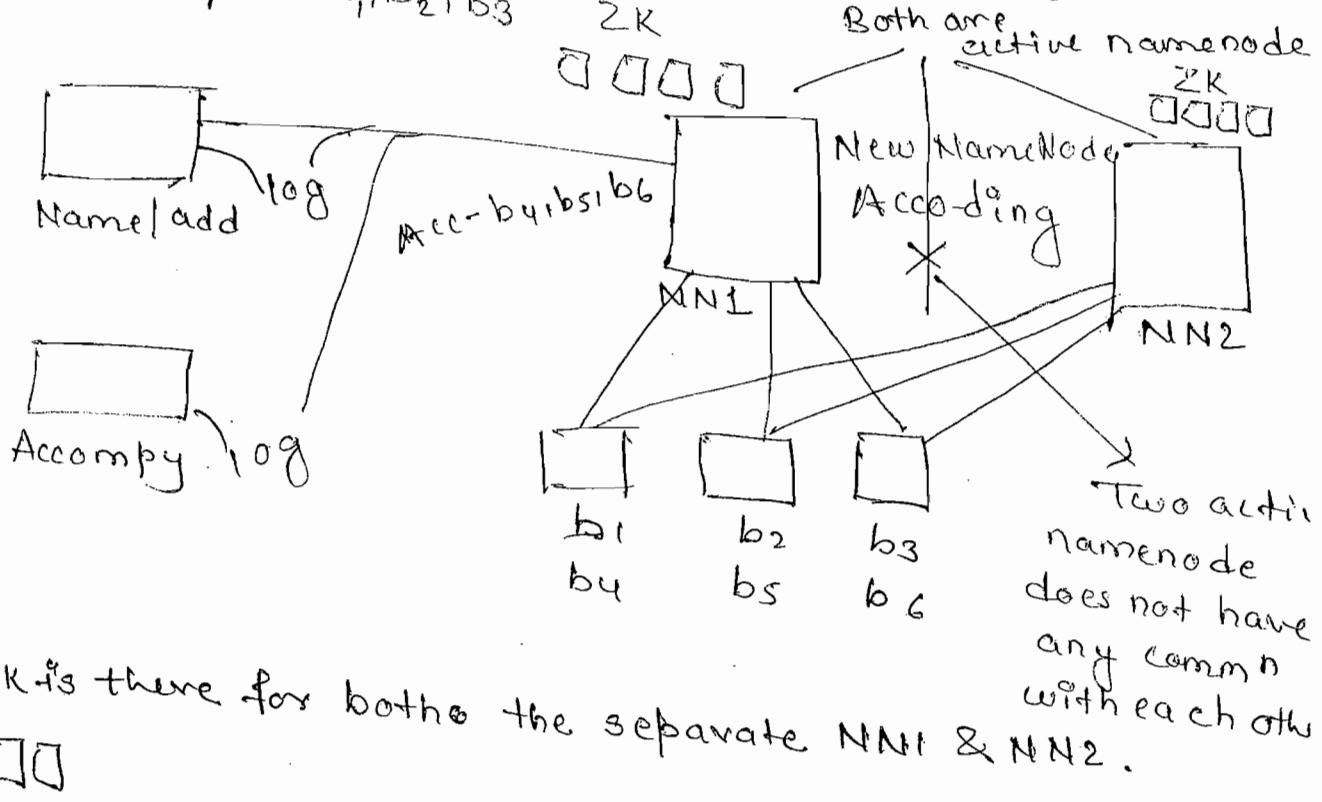
→ If the active node changes, then the client again go to the zookeeper for active node.

HDFS Federation

HDFS Federation

→ Why we have to go for HDFS federation?

NFA → b_1, b_2, b_3



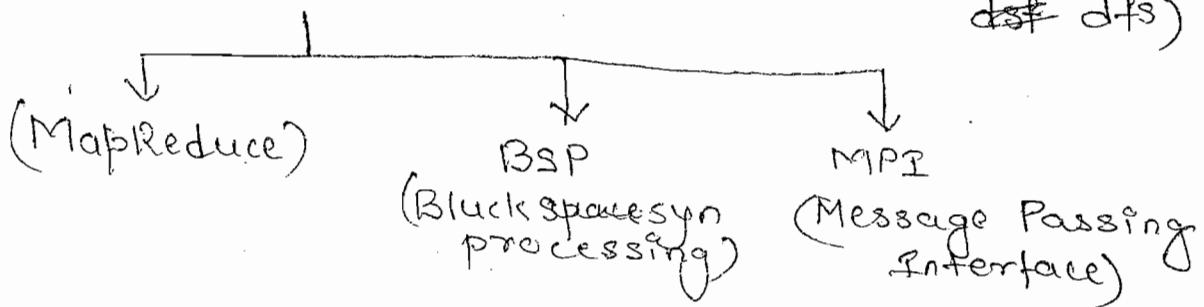
→ ZK is there for both the separate NN1 & NN2.

→ ZK carry the block for high availability.

07/2015
Wednesday

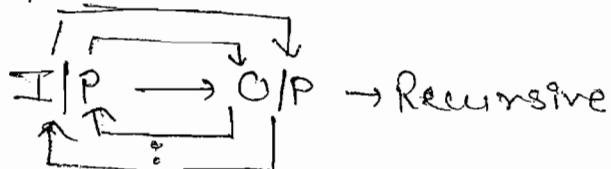
MapReduce -

There are some computational model - (works on
dft & dft)



MapReduce -

- MapReduce is used in aggregation.
- MapReduce is not good for iterative (recursive) algorithm.
eg → ~~map~~, recursive



Bulk Synchronisation processing -

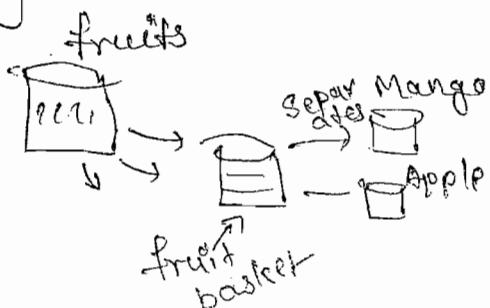
- There is some process called
 - SPARK
 - HAMA
- Iterative Algorithm

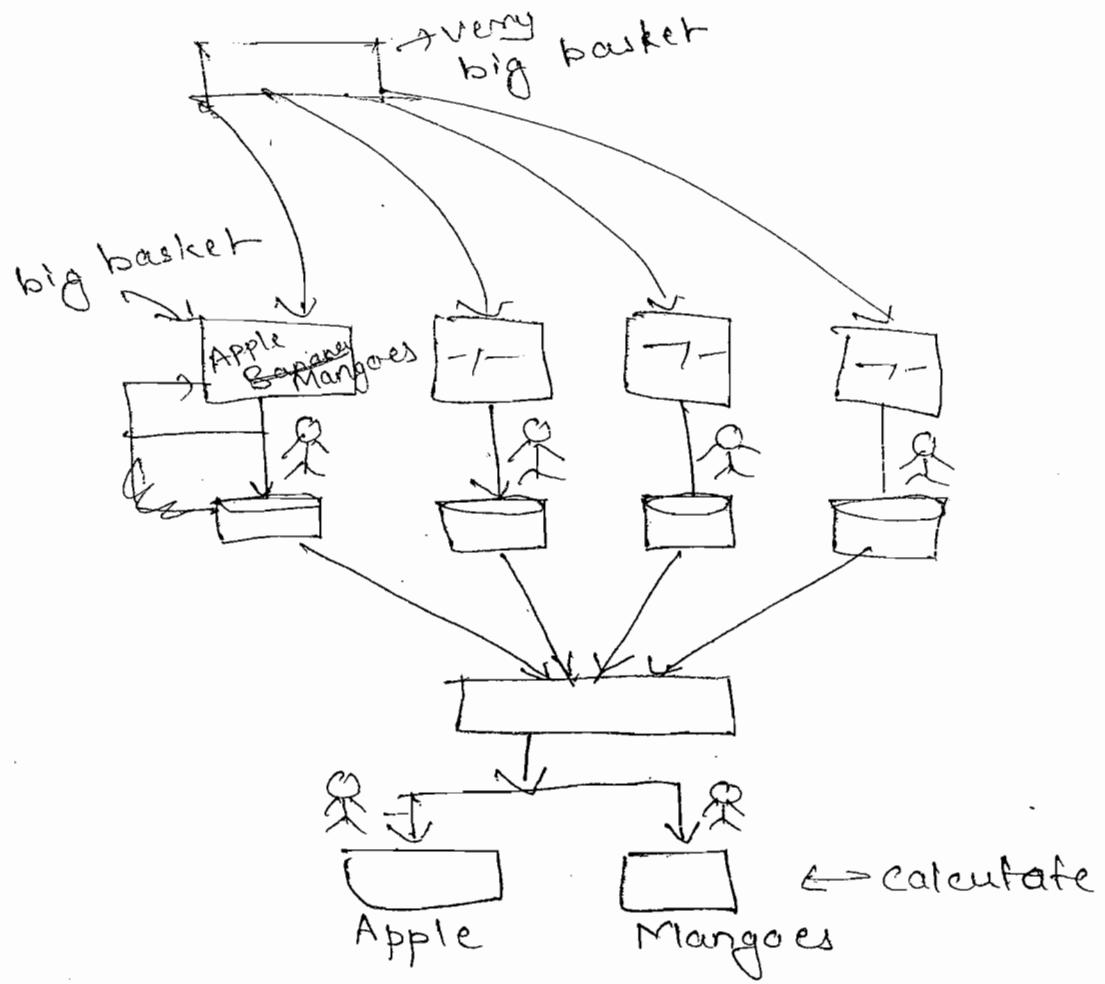
Message Passing Interface -

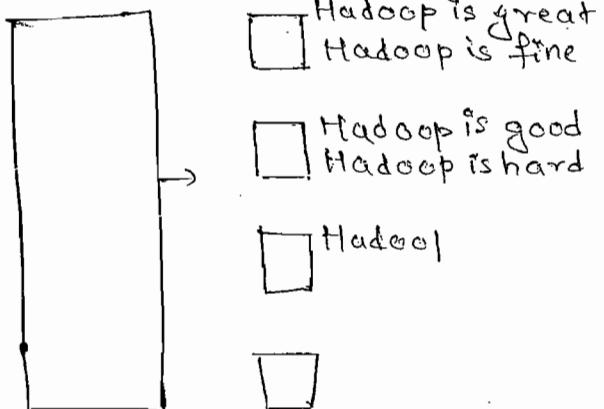
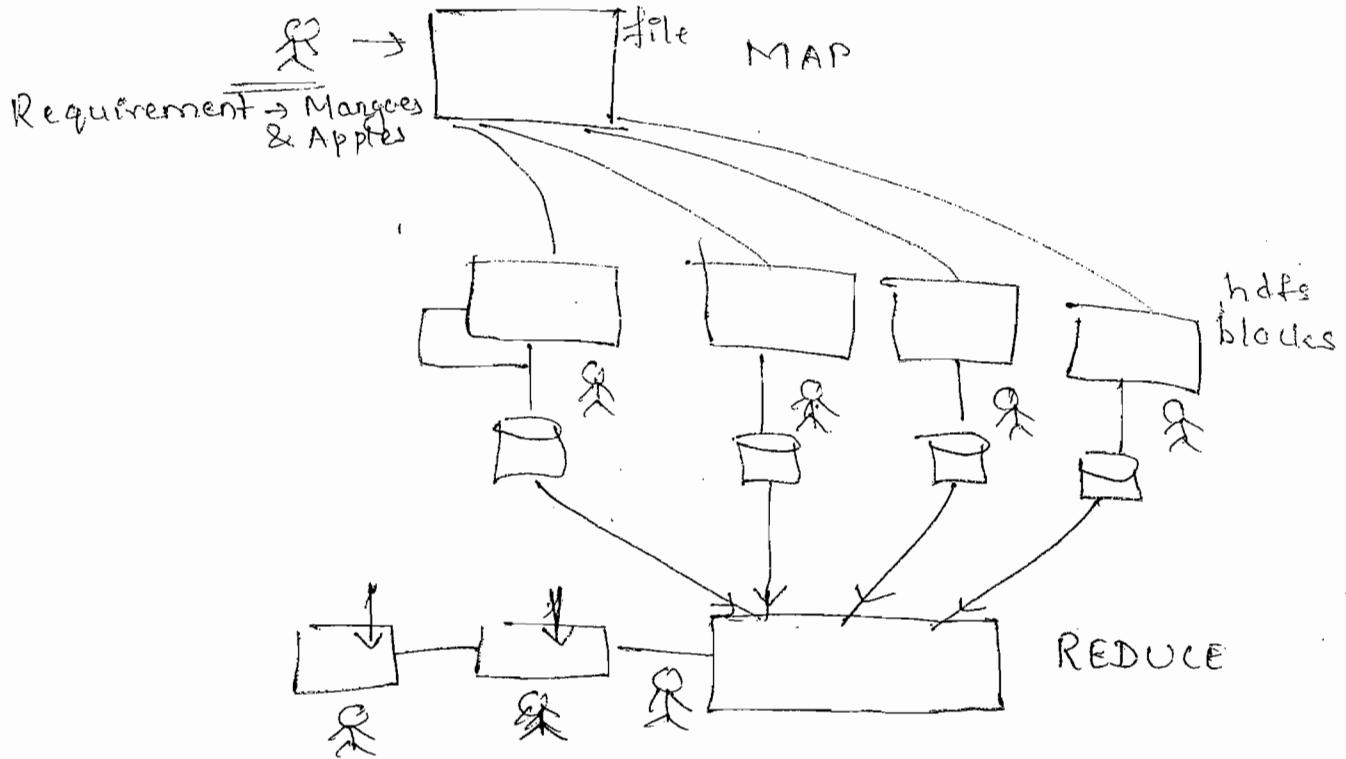
- OPENMPI → ~~X~~ → we do not use it

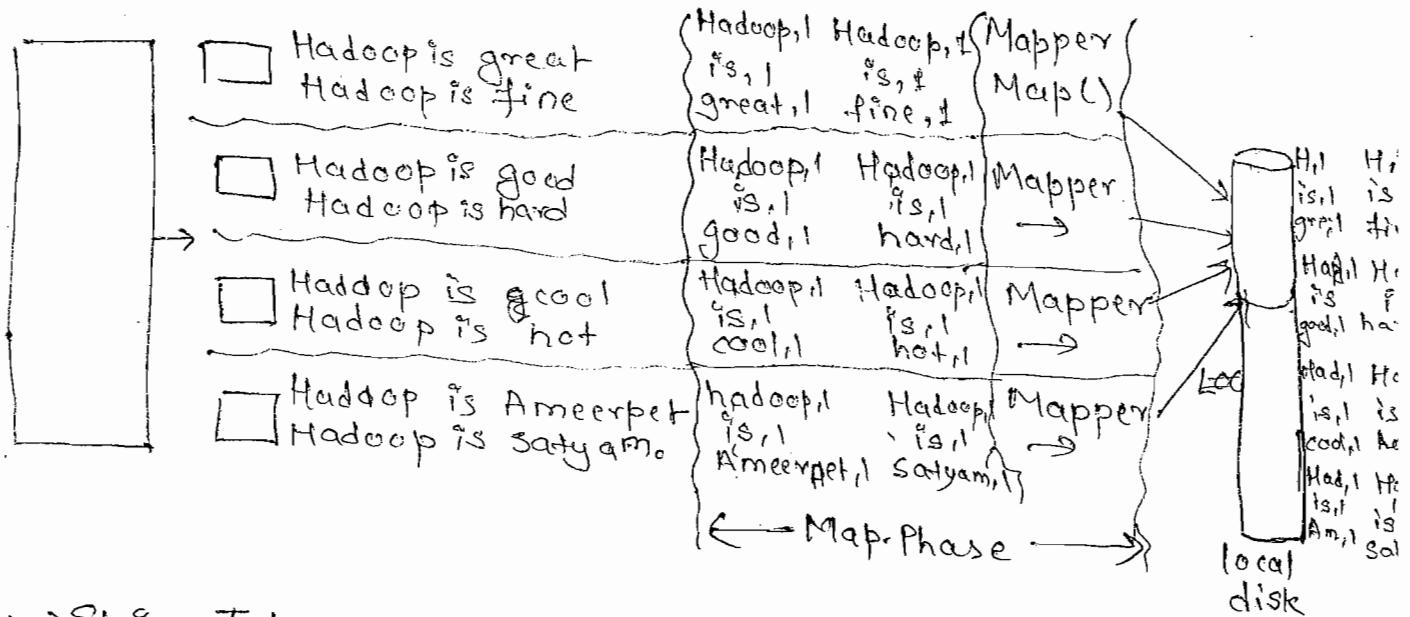
MAP	REDUCE
MAP	FOLD → Functional page <small>(scala, lisp)</small>

- MAP is nothing but transformation.
Ex → group by







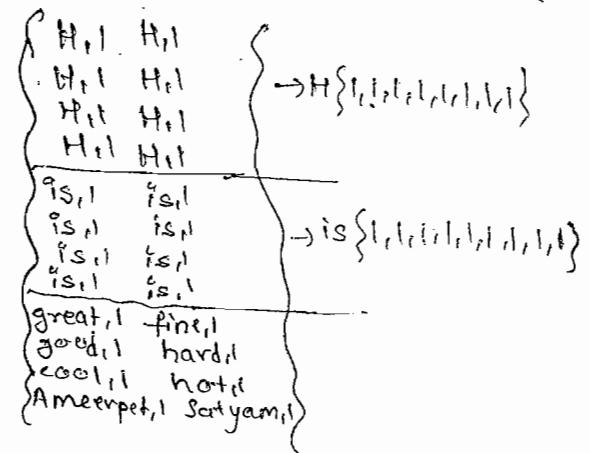


→ String Tokenizer (" \t ")

Hadoop Hadoop
is is
great fine

itr=Stringtokenizer (" \t ");

itr[0],1
itr[1],1
itr[2],1



→ MapReduce gives class to you i.e MAPPER class.

→ No. of Mappers equal to no. of blocks.

No. of Mapper = no. of
^{BT}

= No. of Blocks = no. of Mapper

= Mapper class contains method called Map method, which will be called for every line

Map()

→ Mapper is one of the main class in the MR.

→ It has a lot of predefined methods

One of the methods is map method

This method is called for every line, so whenever any logic related to record level (line level) so we have to write in map method.

- By using Mapper class, we don't need to have locaⁿ of block.
- We don't need to open the file.
- We don't need to iterate the file
- We don't need to close it
- The iteration of MP is allowed developer to concentrate on business logic not on set up.
- Mapper executes parallelly.
- Map() —————— sequentially in each mapper.

→ The O/P of the Map phase will stored in the local disk.

Why the map() ^{O/P}, not stored in HDFS?

- Bcoz Map O/P is an intermediate O/P, which needs to delete after the prog. executn.
- If we stored in HDFS, it will be replicated data.


 local
disk
never
say
down

→ The process of loading data from cluster to local disk is called ~~shuffling~~ Shuffling.

→ The process of bringing like piece together is called Sorting.

→ Reducer is another class.

Which have a method called Reduc() method.

→ It is called for every key level.

Key
S
J
Key 98
H 98 good 1

Reducer -

- Reducer is another class provided by the map - Reduce method.
- This Reducer will going to work after 100% completion of map phase.
- This Reducer have a method called reduce , which will called for key level.
- User have control on the no. of reducers.
- The O/P of reducer is stored in HDFS.

key

{ 1+1+1 ... }

{
8

Key, 8
H, 8
is, 8
good, 8
great, 1
fine, 1
cool, 1
hard, 1

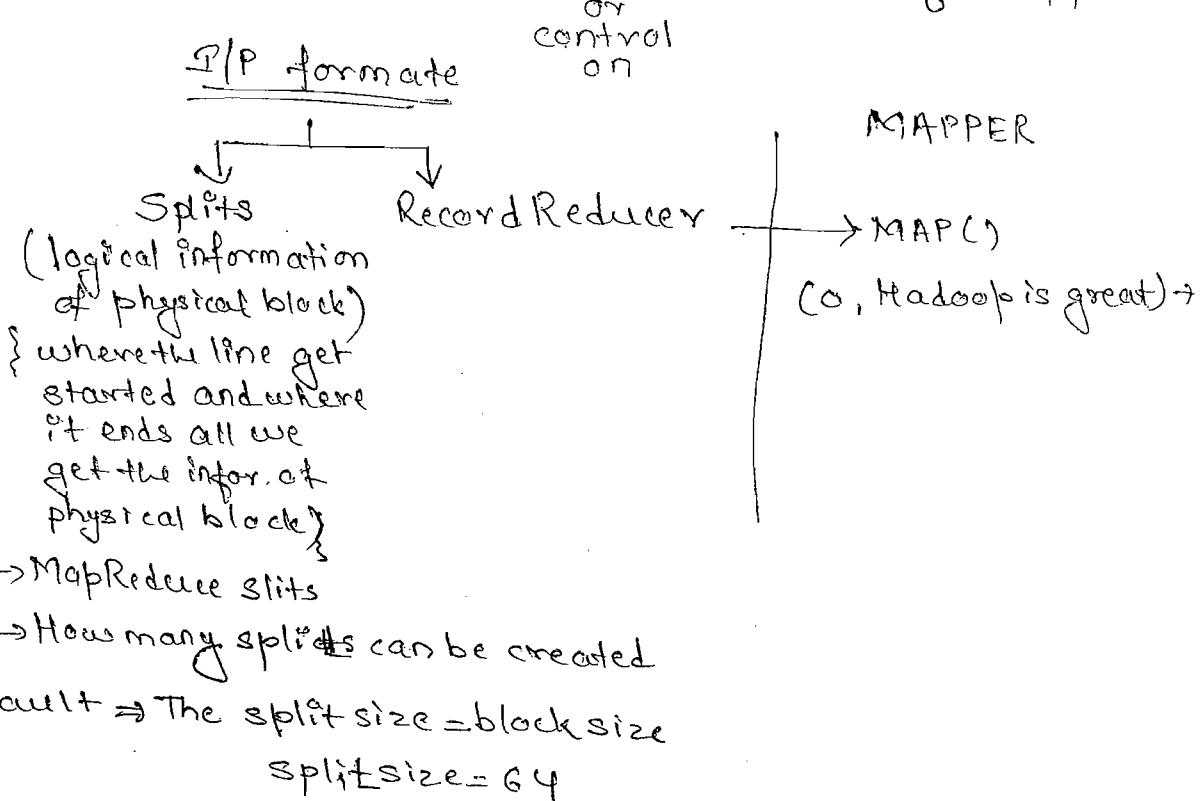
→ The O/P of reducer is always ^{Sorted} in ~~order~~ order.

07/2015
Thursday

~~Default~~ Default -

No. of blocks = No. of Mappers

- It is valid for some file and some file not.
- So the I/P file format decide the No. of Mappers.



No. of Mapper = No. of Splits

No. of Splits = No. of blocks(size)

No. of Mapper = No. of blocks(size)

If we want to increase the time

- Whenever we process any HDFS file, the I/P file format play a very imp role.
- The default I/P processing of your I/P is text file format.
- So the default file format

No. of blocks = No. of Mapper

→ Your tip formate consists of two main classes

(1) → Splits

(2) → Record Reducer

What is I/P split?

→ I/P split nothing but a logical info. of your physical block.

Where the offset started

What is file name.

What is block name

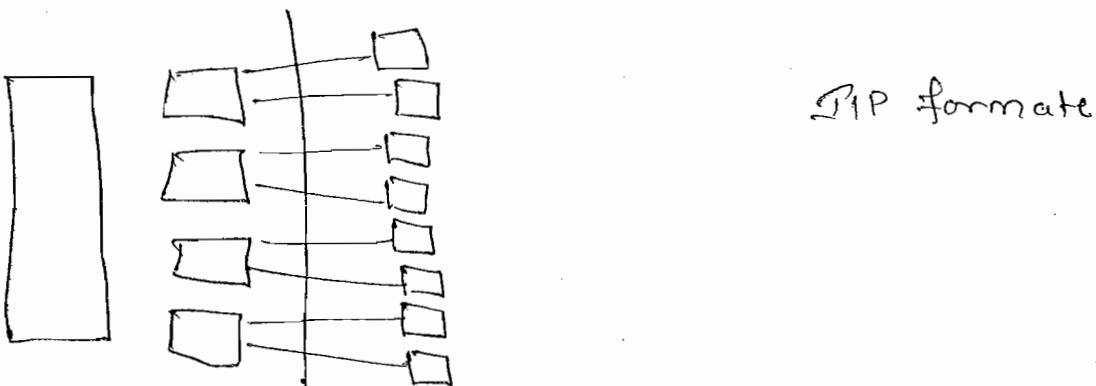
Default -

No. of splits = No. of blocks

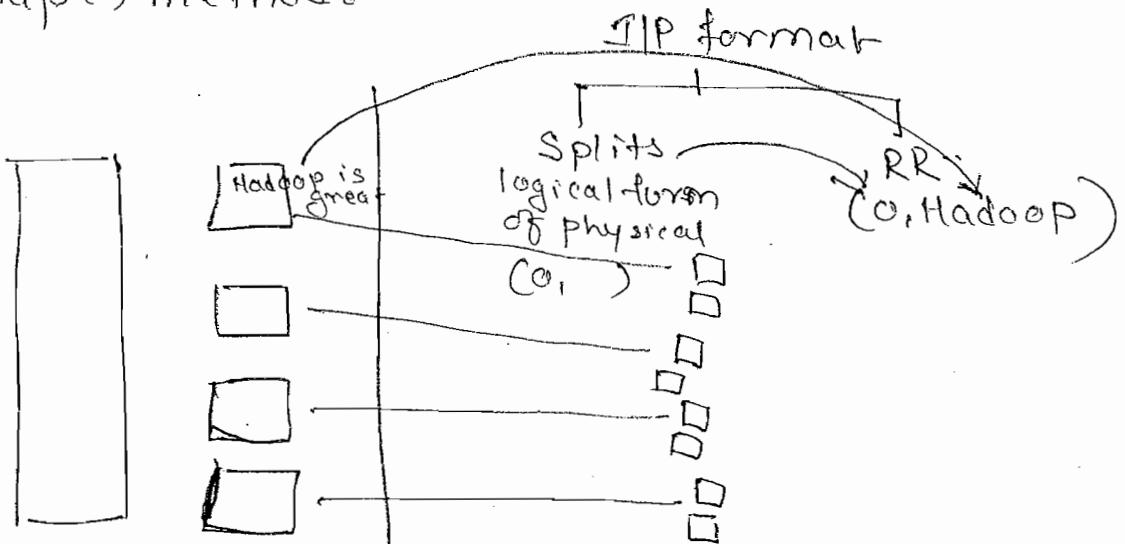
This splits will decide no. of mapper

If you want to increase no. of mapper without functioning.

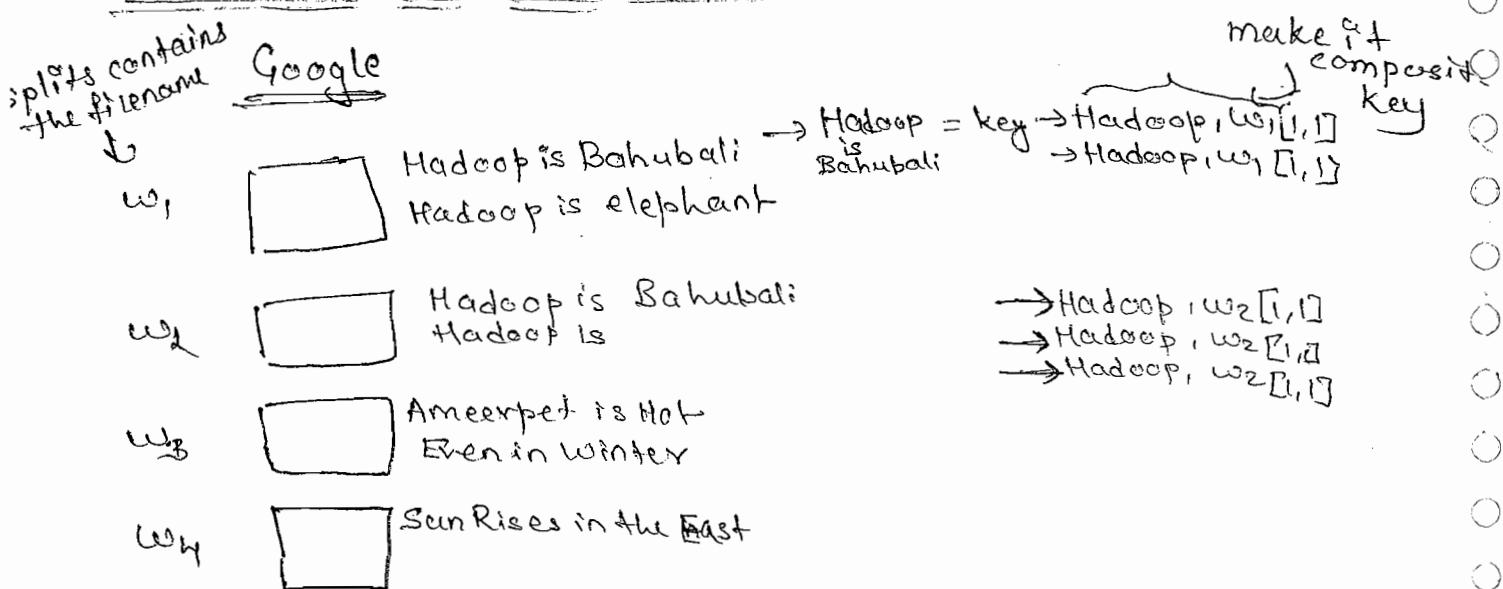
⇒ We have to change the no. of splits.



→ The RR will combine the logical inform and physical inform of blk send it as a line to map() method.



Daemons of our MapReduce -



Marketing Example US

48	85	AP
50	86	TG
50	87	AP
56	88	AP
47	89	TG
44	90	MH
42	91	MH
51	92	MH
41		

→ w₂ is the highest i.e. 3

AP(1)	AP(1,1) - AP(3)	Product Requirement
AP(1)		
AP(1)		
TG(1)	TG(1,1) - TG(2)	
TG(1)		
MH(1)	MH(1,1) - MH(3)	
MH(1)		
MH(1)		

3/07/2014

FRIDAY

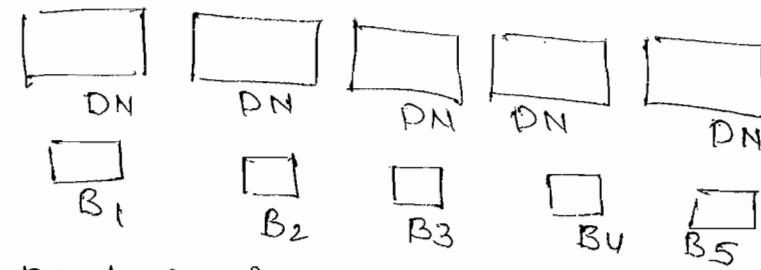
Lifecycle

OS

JT

① Application Mgmt

② Resource Allocation

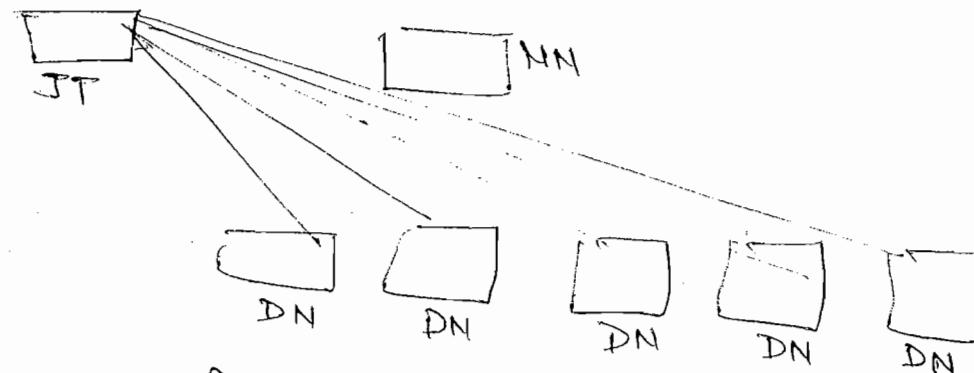


→ The processing part of job mgmt is done by JT.

→ Entire job will be controlled by the Job Tracker.

→ NN is the master of storage data

→ Who will track the job is JT



→ The Daemon of mapReduce -

① Job Tracker →

→ The daemon which have the control over the entire life cycle of the job is called JT.

→ JT has two roles -

① Application Mgmt →

→ Application no which manages the different

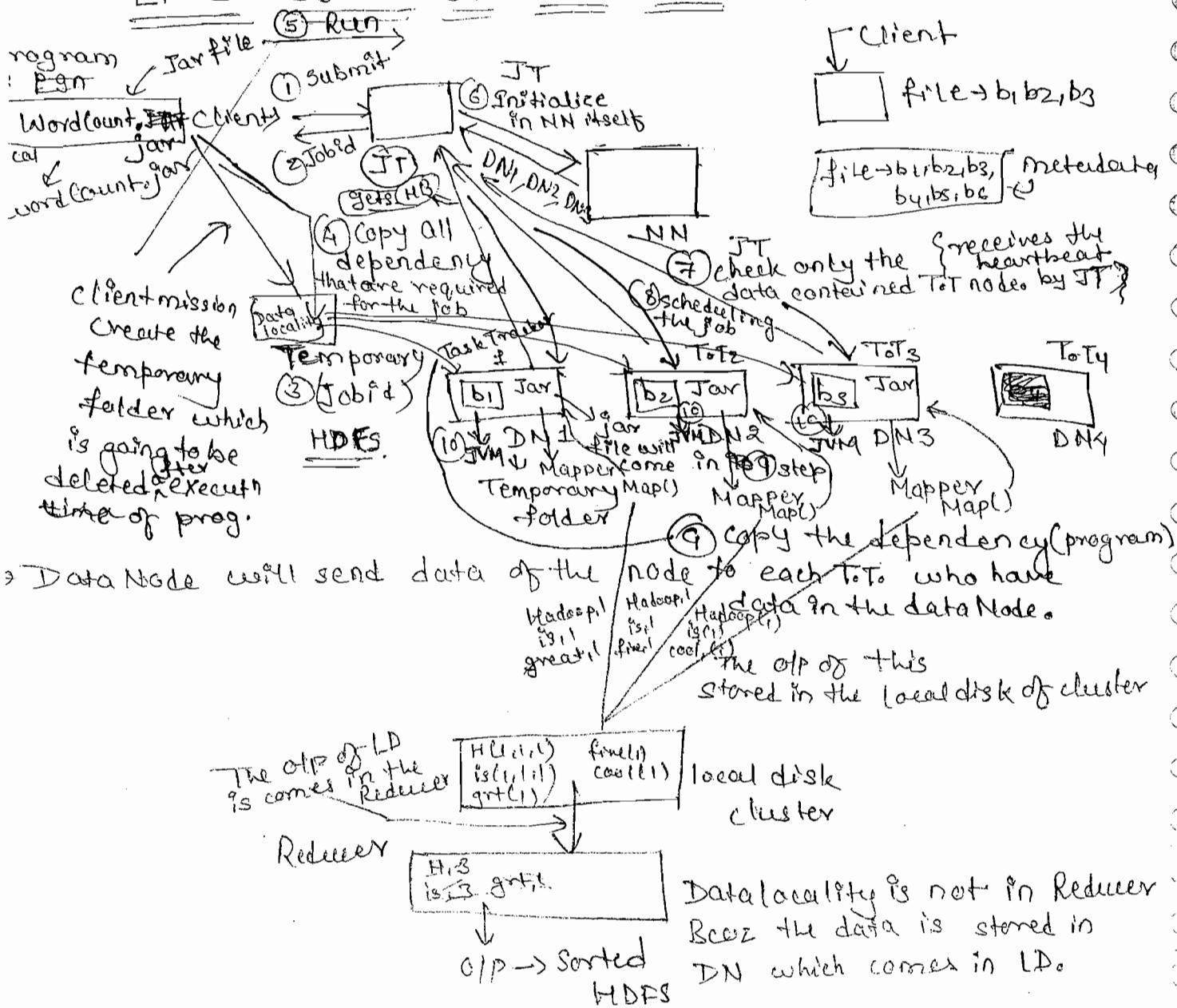
② Resource Allocation →

→ It also act as a resource allocator which allocates required capacity of cluster to perform job.

Task Tracker -

- is a slave of mapReduce
- your job divided into tasks, and each task act as a slave of the mapReduce.
- So, the actual processing is done in the Tasktracker only.

LIFE CYCLE OF THE JOB -



→ Data Node will send data of the node to each T.T. who have data in the data Node.

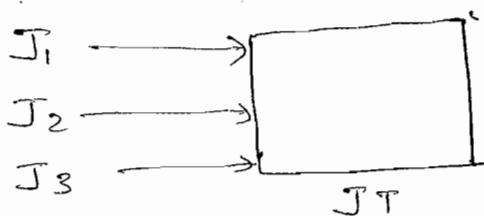
The output of this stored in the local disk of cluster

Data locality is not in Reducer
Bcz the data is stored in DN which comes in LD.

→ Reducer is also controlled by TT.

→ job is to reduce the Map

3 → Types of Scheduler - How we are allocating Jobs



(1) FIFO → (default Scheduler)

→ FIFO is the scheduler which sends the resource from the first to second user.

By which resource gets affected.

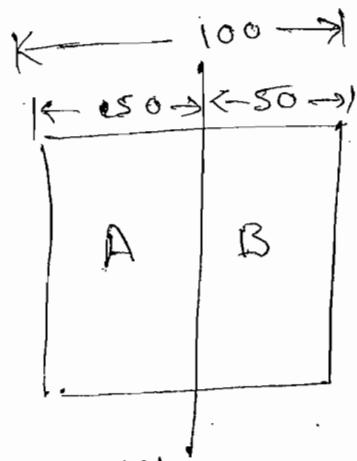


FIFO

FIFO has also some job 1st it finish their own then it goes to send to the receiver.

(2) FAIR SCHEDULER -

→ FAIR shares the job to another scheduler to complete job quickly.



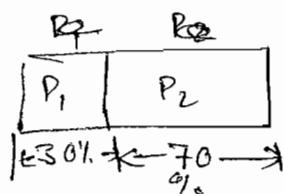
→ If A comes 1st,

→ if shares is having 100%, then B come, it shares 50-50

(3) CAPACITY - SCHEDULER -

P_1
30%.

P_2
70%.



THERE ARE THREE TYPES OF MODES -

- 1) Pseudo Distributed Mode -
- 2) Fully Distributed Mode
- 3) Stand Alone Mode -

STAND-ALONE MODE -

- No configuration file
- Means No Hadoop environment.
- Programs execute sequentially.

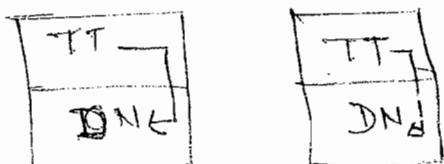
ECLIPSE -

- It is only ^{used} to test the logic not the envt issues.
↓ Deploy in a

PSEUDO-DISTRIBUTED MODE -

- All daemons runs on a single JVM.
- We have config file.
- Hadoop envt too.
- Program executes parallelly. (Parallel Processing)
↓ Deploy in a

FULLY DISTRIBUTED MODE -



~~6/07/2015~~
~~Monday~~

Pseudo Distributed Mode

Don't need to open Hadoop
\$ jps ↴

Open the → eclipse

↳ project explorer

↳ WordCount.java

↳ Src

↳ org

↳ apache

↳ hadoop

↳ example

↳ WordCount.java

Step-1 Create a project in eclipse

Go to file

↳ New

↳ Project



↳ java project ↴ enter

Next

Project Name: WordCount_Mor

Finish

* Step-2 Src, Input

Home

↳ workspace

↳ Bsgdatei

↳ WordCount_Mor/WordCount

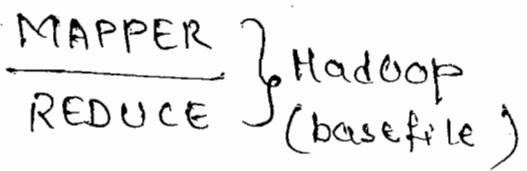
→ ↳ Src from wordCount ↳ input

copy the input folder, into it

Now open WordCount_Mor

→ you can see input folder

Now, we going to write program in java



Step-3 - Load the dependency (Add jar file to project)

Go to the Installations

- ↳ Hadoop 1.2.1
- ↳ lib

Go to the WordCount_Mor

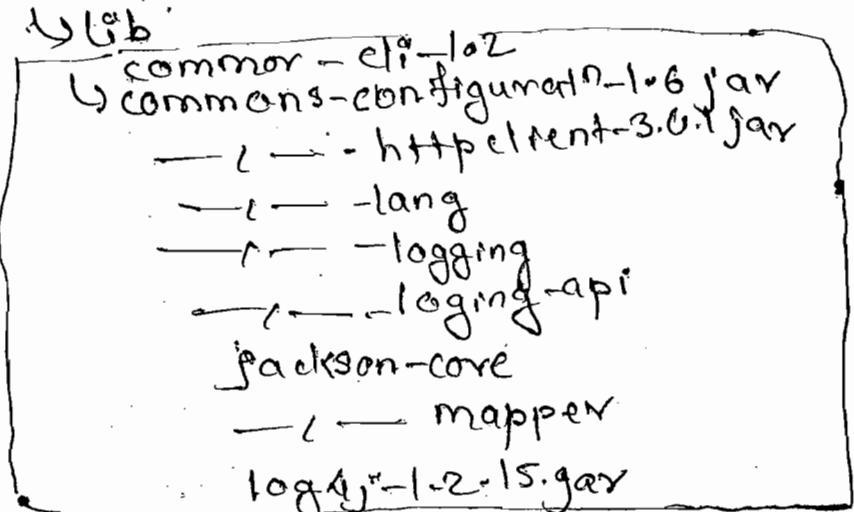
- ↳ jar files ↳ which we need to add in our project.

WordCount_Mor

- ↳ Java Build Path
- ↳ Configure Build Path
- ↳ Add external jarfiles
- ↳ Installation

↳ right click

↳ Hadoop-1.2.1



Say → external jar again -

- ↳ hadoop-core-1.2.1.jar

right click on src

we will going to create a class

↳ New

↳ Class

④ Create the class → src

src

↳ New

↳ Class

Name: WordCount

Finish

WordCount.java
- public class WordCount

Now delete WordCount

src

↳ WordCount.java org.apache.hadoop.example

↳ copy

WordCount.java / src → paste

Divide into three parts

- ① → Driver class (main class is the driver class)
- ② → Mapper class
- ③ → Reducer class

Now we are going to talk about
Driver class first.

src

↳ Driver →

main(

{

args[])

↳ IP -> HDFS
OP -> file only
so the
main method
get two
parameters

main(args[])

{

① Configuration class object

Every Hadoop class consist of 3 main prog.

- ① Driver → where actual process ^{control} get started
- ② mapper
- ③ reducer

main(args[])

{

① configuration object. [xml files]

for ex → conf

↳ you created

→ it very depend on
modes

→ create the obj. to run

the mode.

② Creating the job [conf]

→ job is connected with the
new conf file.

identity
Mapper ↳ ③ Job. Set mapper (WordCountMap.) → wordMap class

↳ give the class of the
mapper

If you don't specify, the default job will call.

Default Mappers —

A mapper which will call for every line,
that type of mapper is called identity mapper.

(4) → Job.setMapperClass(WordCount.class) —

→ Give the ZIP file folder

Default file format → txt file

(5) - Job.waitForCompletion(true);

Goto

WordCount.java, sample.txt | WordCount.java
↳

Mapper

TokenizedMapper extends MAPPER

⇒ You should not use Java datatypes in Hadoop
bcz Java datatypes are not serializable.

convert the datatypes into
byte & that bytes are
again convert into the
respective language.

⇒ It will run on JVM but not serializable.

→ Writable package is default package in Hadoop
just like lang → in JAVA.

→ Writable Comparable // base class

Datatypes

JAVA

Hadoop

INT



IntWritable

Float



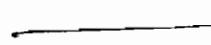
FloatWritable

double



DoubleWritable

String



Text

⇒ TokenizerMapper extends Mapper

(Object, Text, Text+, INITIALIZABLE)

⇒ MAP (Object key, Text value)

Mapper ⇒ Hadoop framework work is called Context.

Q, Hadoop is easy
NT

↓ string	Hadoop, is, easy,)	↓ Text	K, M → List (K ₁ , V ₂)

0 → is the starting position of object

Hadoop is easy → text

Hadoop
is
easy } → text

| } initwriter

```

public class wordCount {
    private final static Text writable one = new Text();
    ↑
    const
    ↑
    variable
    ↑
    value
    private Text word = new Text();
    ↑   ↑
    Variable value
}

```

```

public void map(Object key, Text value, context context)
    throws IOException, InterruptedException {
}

```

```

System.out.println("→ In the mapper. line = " +
    value.toString());
}

```

Hadoop is easy

itr ⇒ Hr[i] = HADOOP

itr[j] = is

itr[t] = EASY

```

StringTokenizer itr = new StringTokenizer(value,
    " ");
}

```

```

while (itr.hasMoreTokens()) {
}

```

```

    word.set(itr.nextToken(),
    context.write(word, one));
}

```

$\left\{ \begin{array}{l} \text{Hadoop}, 1 \\ \text{is}, 1 \\ \text{Easy}, 1 \end{array} \right\}$

Now we have to write some coding lines -

```

System.out.println(key + " " + value.toString() + " "
    word + " " + one);
}

```

Hadoop	1
is	1
Easy	1

context

?

Now for Reducer class

public static class IntSumReducer extends
Reducer<Text, IntWritable, Text, IntWritable>

The O/P of map datatype will become I/P for the
reduce.

(k_1, v_1)
 $\hookrightarrow \text{list}(k_2, v_2) \rightarrow k_2, \text{list} v_2 \rightarrow \text{list}(k_3, v_3)$

Hadoop, (1, 1, 1)
is, (1, 1, 1)
easy, (1)
hard, (1)
cool, (1)

$k_1, v_1 \rightarrow \text{Map} \rightarrow \text{list}(k_2, v_2)$

reduce
 $k_2, \text{list} v_2 \rightarrow \text{list}(k_3, v_3)$

$k_3 \rightarrow \text{Key}$
 $v_3 \rightarrow \text{result}$

private IntWritable result = new IntWritable();

input

sample.txt → Property

wordcount.java

↳ Run As

↳ Java Application
Run Configuration

Select - project

Main.class: WordCount_Main

org .. -

Now go to Arguments

program argument

homelum4\learning\workspace\bigr data /sampletxt

~~Input~~ Output

FINISH

Now you can see the α 's easily...

Output →

Part-r-00000

Hadoop-3

cool = 1

easy - 1

Cast, 1

$$r = 3$$

Copy internet page

and copy into
Hadoop
8.2.1

find which word
repeated many
times.

07/2015
Tuesday

Steps

- ① Start the cluster → Start-all.sh
 - ② Place files into hadoop cluster → i.e. HDFS
 - ③ Format the cluster
- Home
↳ PublicDataSets
↳ Gutenberg
↳ Goto property → Select the location
Start the cluster

bin/hadoop fs - put /home/vm4learning/Public
DataSets/Gutenberg /user/vm4learning/input/

— Place files into the cluster

- ④ Job tracker will take everything in jar file

\$ bin/hadoop Jar

Create a jar file from the project —

→ Right click on word-count-Mor

- Select Export
- jar file

Jar Export:

Select resource

Word Count

Finish

WEB CONSOLE → localhost 50030
for Job
Tracker

\$ bin/hadoop jar <your_JAR_LOCATION>
↳ in Mission

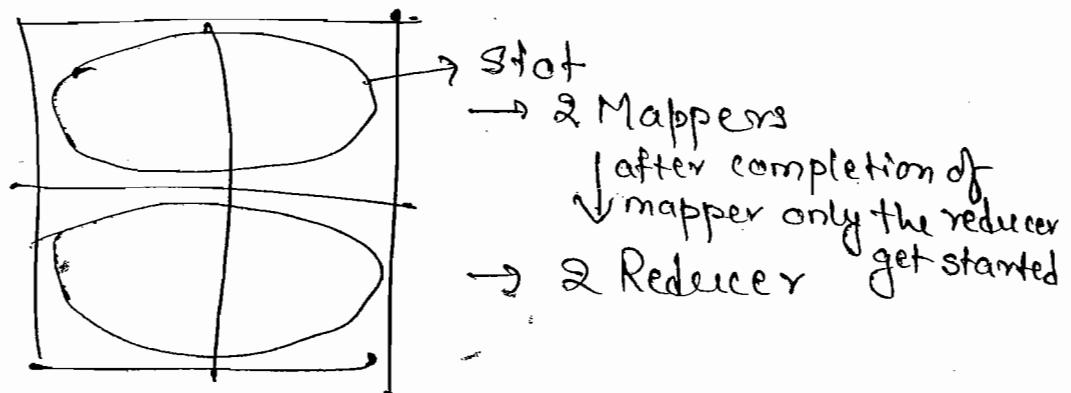
<MAIN-CLASS>

<IP>

<OP>

This two IP & OP
class is input for
Main-Class

\$ bin/hadoop jar /home/vm4learning/Desktop/
WordCount.jar org.apache.hadoop.examples.
WordCount /user/vm4learning/input3/usr/vm4learnin
/output.



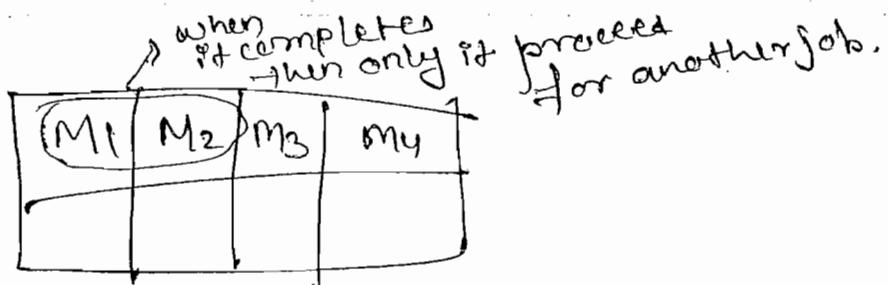
Default is → 2 Mapper slots

But we can change in 4 slots

For 4 → Mapper

4 → Reducer

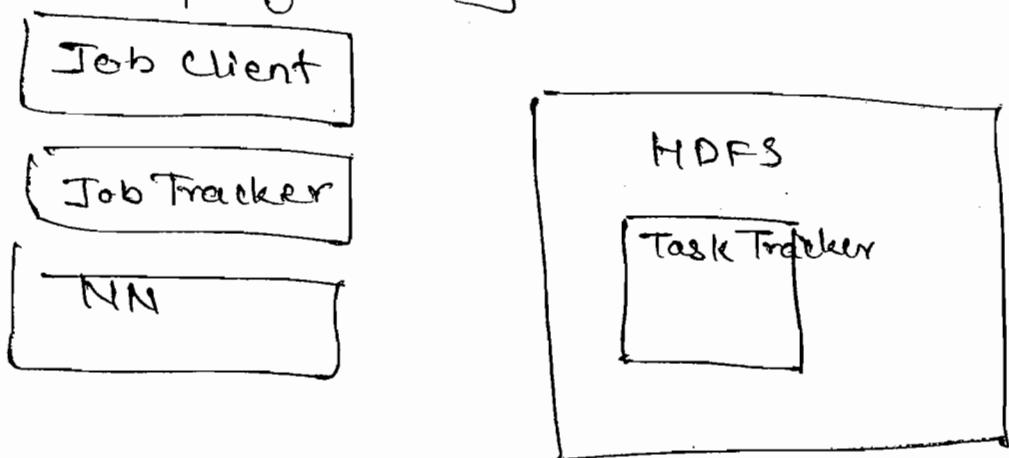
Job 1 = 3
Job 2 = 3



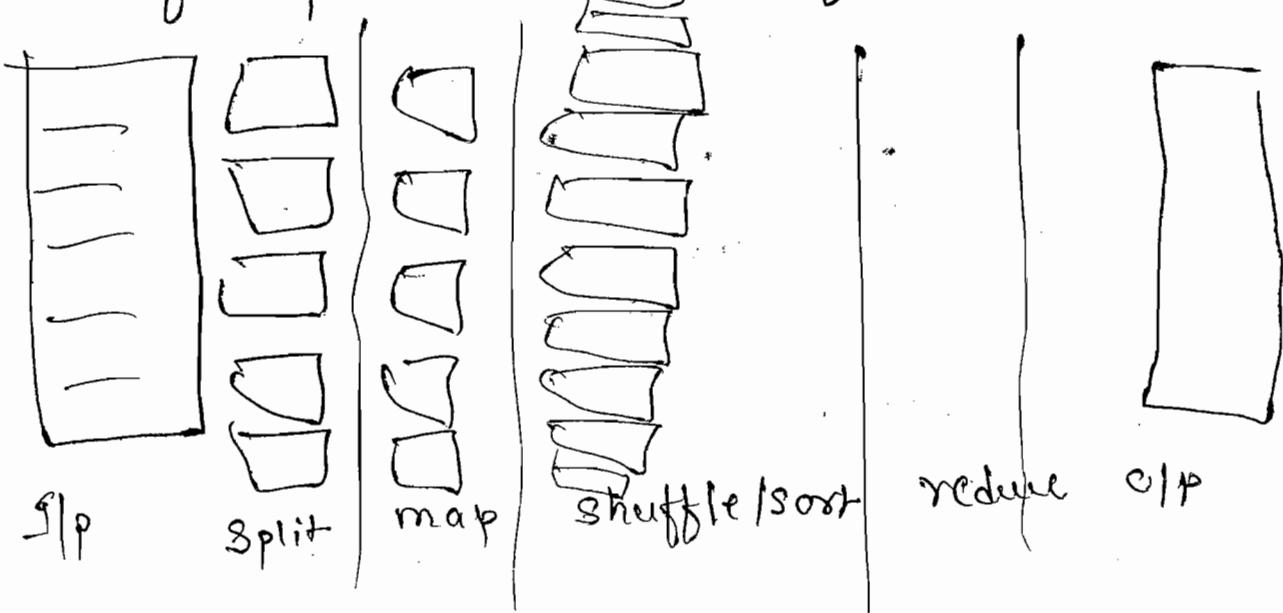
To check the fifo order —

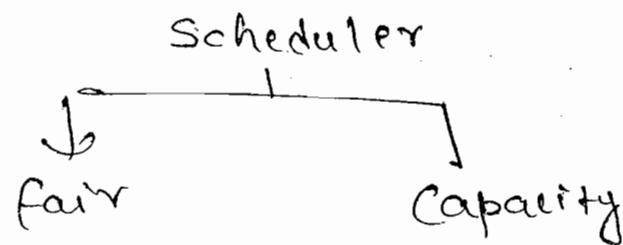
Submitted the 2 job parallelly

In the same way, we can put multiple jobs.
See the progress in jobtracker web console.



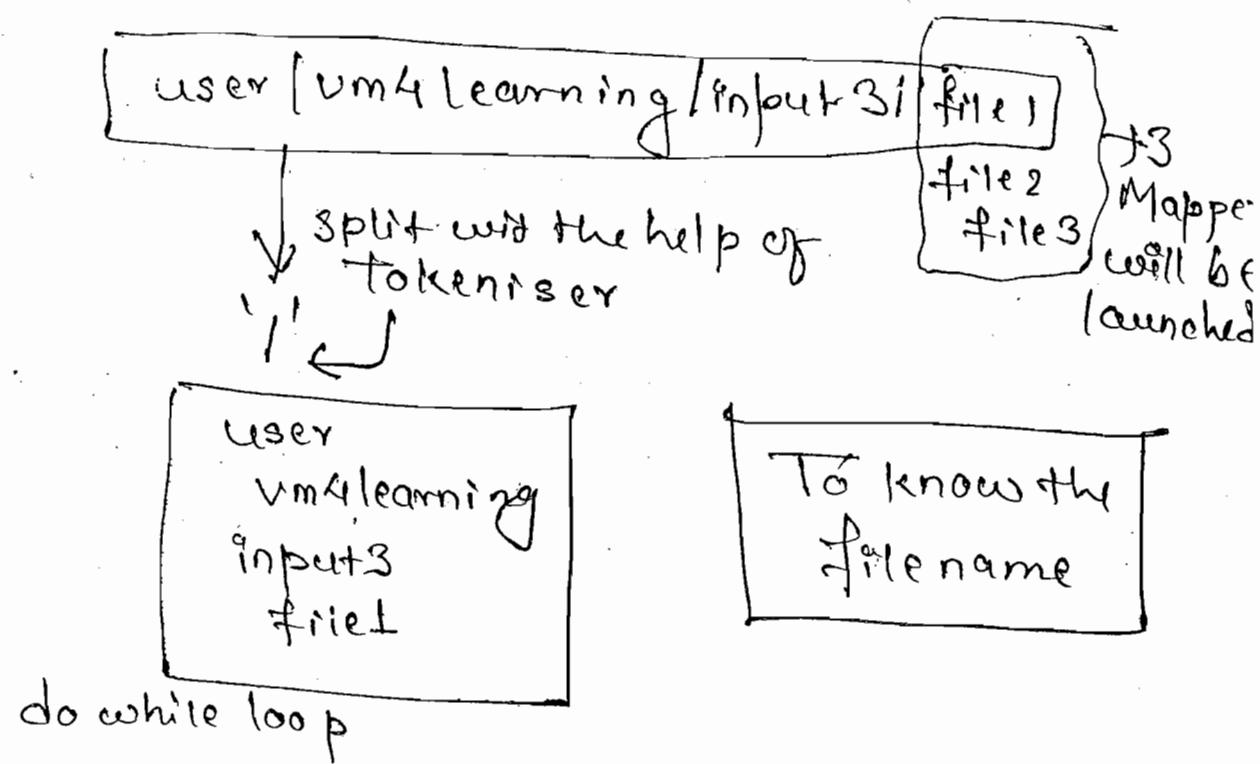
No. of splits equal to No. of Mapper.





→ Goto the Web Console

⇒ "the" contain in every file repeat button which file "The" repeated from 3 file.



File.name → File \$

(file \$, word, l)

where will be write this logic \rightarrow Map() method

⇒ It will execute for every ~~time~~^{line} \Rightarrow Disadvantage
Before MAPC() method

The mappers has many method in class -

③ Map() method -

→ will execute for every line.

If any method wants to execute only once in the life cycle of mapReduce before Map(), for this kind of situation , we have Setup().

→ This 'Setup()' will be called only one time in the lifecycle of MapReduce before Map().

→ ~~other~~ The method is called CleanUp() which is called after map() in the entire lifecycle of the MapReduce.

8/07/2015
Wednesday

SETUP() → call only one time in the MapReduce entire life cycle.

MAP() → calls only one time

CLEANUP() → calls only one time

(1) Cloudera → Hadoop User Env

Already built, we do not need to anything in this.

(2) HDP

—x—

Open WordCountPerBook

& Open Eclipse

Hadoop is Great → O, filename

Input string → schema → not contain logical path

control f say the

REDUCER -

(3) types of Reducer -

① Identity Reducer (Default) -

job.setReducer(f); doing some aggregation
 ↳ override the Reducer()

	Sorting	Aggregation
Identity key	✓	✗
zero Reducer {}	✓	✓

Job.setreduce(0);

zero Reducer {}	✗	✗
-----------------------	---	---

BenchMark Testing
Defaultly & Identity
with all.
make it in sorted
order.
Text → seq.

~~12015~~
~~Thursday~~

Why we need to go for one more than
one & reduce.

- To perform our reduce opⁿ, ^{parallelly} we can set more than one reducer.
When we place more than 1 reducer, we will going to have better readability (to get opⁿ more efficiently) which effects the performance.

① Lookup.txt

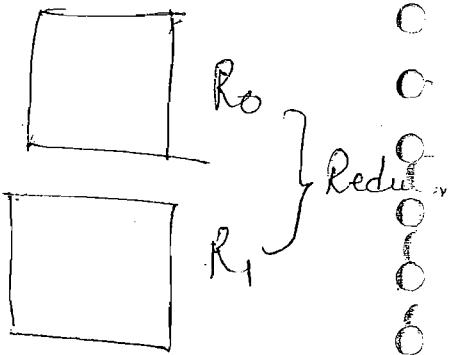
RegionId	Region	RegionState
45	Andhra	→ SI
46	TG	→ SI
47	Chennai	→ SI
48	Delhi	→ NI
49	Punjab	→ NI

Create log.txt

User Id	Product ID	Region Id
101	100001	48
102	100002	49
103	100003	51
104	100008	46

we will emit
to

key	value
48(101,10001)	
49(102,10002)	



- There is a problem arises of rooting the reducer.
- Partitioning is the other phase of the reduce.
- Partition is one of the state in the mapreduce().
- The fun of partitional is to root the specific key to the specific reducer.
- A partitional is used the default method ~~for~~ called Hash partitioning.

Partition

↓

→ Hash partition
 $\frac{\text{hash(key)}}{\text{No. of reducers}}$

$$\text{hash}(48) \% 2 \rightarrow (0, 1)$$

0

1

→ Hash partition of reducer is going to root the key.

$\boxed{\text{hash(of key)} \% \text{ no.} \rightarrow 0, 1}$

$$\text{hash}(48 \% 2) \rightarrow 0 \rightarrow \boxed{0}$$

$$\text{hash}(48 \% 1) \rightarrow 1 \rightarrow \boxed{1}$$

$$\rightarrow \text{hash \% 1} \rightarrow 0$$

⇒ hash is performed even when there exist only one Reducer.

MAPPER $\xrightarrow{\text{method}} \text{MAP}()$

Reducer $\xrightarrow{\text{method}} \text{Reducer}()$

partition → getpartition()
partition → overriding functionality

0 ↗ { If (key < 15)
return 0
else
return 1 } Range partitioning

0 - 15 → North India
15 - 30 → South India
Hash is having a problem, that's why we're using Range partitioning.

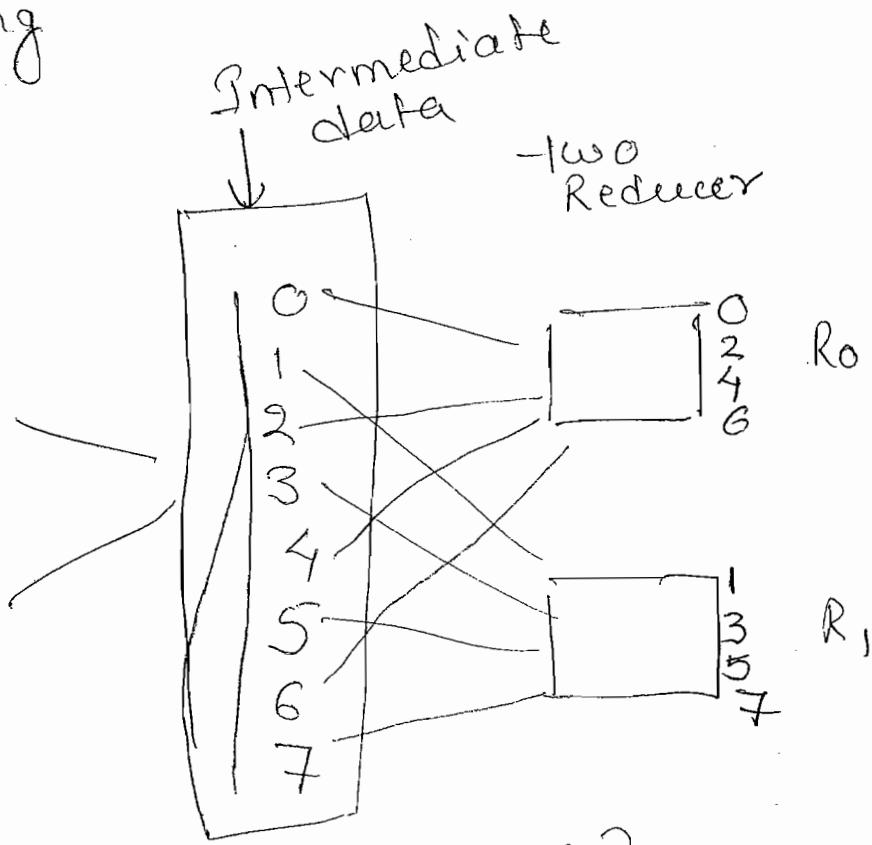
→ Compiling
 ↳ Total Sorting.java
 input → file1
 ↳ file2

→ bin/hadoop fs -put /home/vm4/learning/workspace/Bigdata/input /user/vm4/learning/sorting

⇒ Place the file into a folder from input to sorting

Coding =

2	2
4	4
7	7
6	6
1	1
3	3
0	0
8	8



$$\text{hash}(0) \% 8 = 0$$

$$\text{hash}(1) \% 8 = 1$$

,

,

,

↑

0 < 4 → 0

else

→ 1

0 } R0
1
2
3

4 } R1
5
6
7

→ Map output is called Intermediate data
↓
very huge

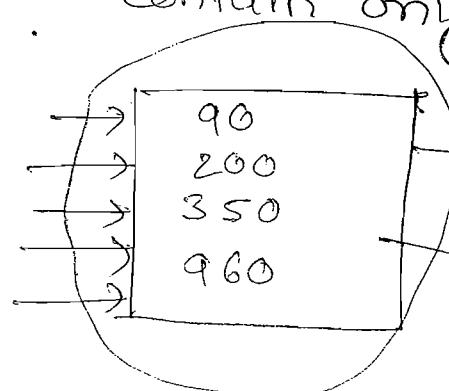
Total Order Partitioning -

Input Sampler

↓
from huge data set

Take randomly the keys

and put into some other data set which
contain only keys



key contain dataset

→ Median (to find the median
value in range partitioning)

→ Input Sampler helps to make task easy
to pick some random keys to find
the median value in range partitioning.

Goto src

↳ TotalSorting.java

Now open → Range Partitioner.java
Control + m

Age level -

1-20 → R₁

20-40 → R₂

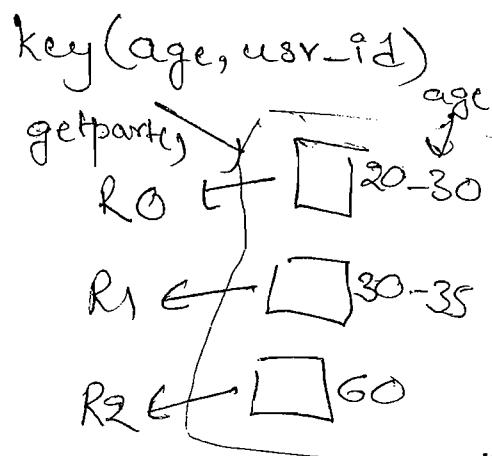
40-60 → R₃

adidas → 20-30

pension → 60

Samees → 30-35

userid	age	country
1	25	India



getpartition ()
if (age >= 20 & age < 30)
return 0;
else
if (age >= 30 & age < 35)
return 1;
else
return 2;

How to Run using Python, Ruby, R
Map Reduce jobs

10/07/2015
Friday

HUE → HADOOP USER ENVIRONMENT

↳ HUE

① Create a twitter account

② ↳ settings

↳ get something like Apps

<http://dev.twitter.com/apps>

How satyamernjayte analysis twitter & increase TRP

④

Create New App ← click on

Create an application

Application Details

Name * TNaresh-Sentimental ← fill the form

Description * []

Website * []

Yes, I agree

[Create]

⑤ Go to Keys & Access Tokens

① Consumer keys
 ② Consumer Secret } very important keys.

⑥ Your Access Token

Create An Access tokens

① Access key

} pls keep in mind

② Access Token Secret } when you are working
on twitter.

1st Method To Get Tweets

⇒ there is a package called **tweepy**

⇒ we are going to download the tweepy
tool from the package.

→ After **download**

↪ → ↪ **tweepy** ↪

↪ **sudo python setup.py install** ↪

Then it will be installed.

open ⇒ **twitter-fetcher.py**

③ Take Consumer key

consumer secret ↪

Access_Token-KEY ↪

Access - Token - SECRET ↪

} Copy
Paste
Karna
hai

Q
C
C - Auto Clouded

C - \$ python /twitter-fetcher.py /Banubali.txt

C - 2nd Method

C - (1) Twitter ↴

HADOOP STREAMING -

C - → Running the MR job w/o Java Code.

C - → Processing framework the data by any language which will emits the data to the I/O streams

C - Java

C - Sum(a+b)

C - Terminator

C - \$ echo "Hi Baby This is Arvind waiting at Satyam
Cinemas come fast Baby" ↴

C - In Home

C - ↳ Code

C - ↳ Streaming

C - ↳ word-count-map.py

C - | word-count-map.py

C - | ↳ std.inp. ↗

[for line in sys.stdin] → |

word_count_map.py
↳ Location copy

cat tweet.txt

↳ to the file

\$ echo "Hi Baby ---" | /home/vm4learning/code/streaming/word_count_map.py

Hi 1
Baby 1

|
|
|
|
|

theater 1

|
come 1
feet 1
Baby 1

\$ echo "Hi Baby ---" | /home/vm4learning/code/streaming/word_count_map.py

--- /word-count-map.py | sort

⇒ All the things come in sorted format

| /home/vm4learning/code/streaming/word-count-reduce.py

Open
wordcountReduceopy

Distributed Cache -

Deploy

Ecumy leaves

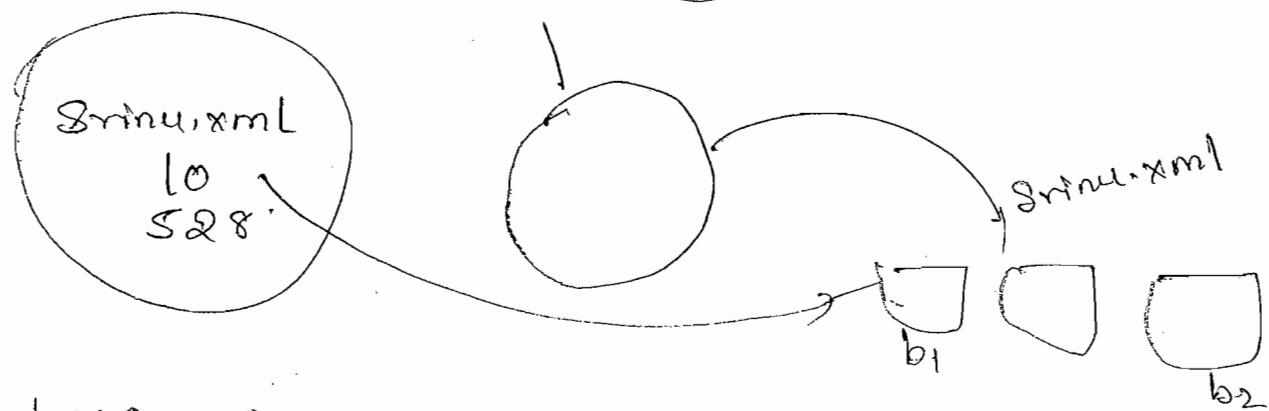


Install the file at the runtime execn and
after execution the file will be deleted

\$bin/hadoop jar project-files Srinu.xml MainClass

CLI ↔ IP

I/O



how Distributed
cache will work?

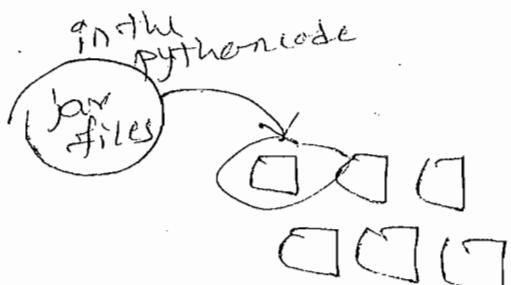
\$bin/hadoop jar --- DataNodes
map & d ← checkout

Prinuoxml used case → poc → proof of concept

→ To convert audio into text

— for this we use

Converter.jar



diff b/w convert A to B

file and libjar

↓
file will
copy to
the src

libjar is file
and takes time to copy
→ set a class path → say
libjar

and then copy to libjar
which take less time.

bin/hadoop jar /home/umakant/

Installation

Hadoop 1.2.1

↳ contrib

↳ Streaming

↳ hadoop-streaming 1.2.1.jar

↳ properties

bin/hadoop jar /home/paste /

bin/hadoop jar /home/vm4learning/Installation
/hadoop-1.2.1/contributed/streaming/hadoop-streamer
→ hadoop.jar -file /home/vm4learning/code/streaming
word_count_map.py -file (1) word_count_reduce.py -file (2) property of Reduce
Home/Code/streaming/set the Map properties
Reducer

→ mapper word_count_map.py
→ reducer word_count_reduce.py
→ input /user/vm4learning/input3
→ output /user/vm4learning/python

Syntax —

bin/hadoop Jar (Jarfile)

→ file

→ file

→ mapper

→ Reducer

→ reducer

→ I/P

→ O/P

Hadoop
Streaming ?

~~Yester~~
~~Saturday~~
13/07/2015
MONDAY

Cloudera Install

File

↳ Import Appliance → select → Next → Import
Next

↳ Desktop
Find Max. Temperature

file:///home/vm4learning/Desktop/vm/Hadoop-

legacy.html

MR LAB1.txt ↵

Go to

↳ Home

↳ Public Datasets

↳ Weather-Single file

↳ Sample.txt → properties - copy the file path

Without starting Hadoop, we do the job

feat [copy the path] ↵

Now we going to Analysis on this huge file -

Go to Home

↳ Code

↳ Streaming

↳ max_temperature_map.rb

```
$ cat /home/vm4learning/Public Datasets/  
weather-singlefile/Sample.txt | /home/vm4learning/  
Code/Streaming/max_temperature_map.rb | sort
```

max-temperature-reduce.rb

↳ properties → copy the file path

\$ cat --- | sort | /home/vm4learning/
code/streaming/max-temperature-reduce.rb

Now start your hadoop cluster

\$ bin/start-all.sh

Open

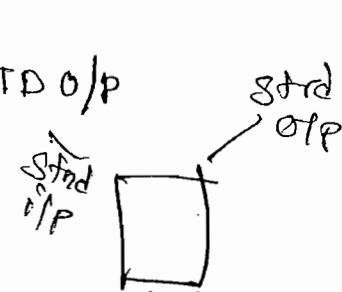
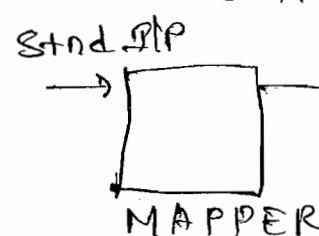
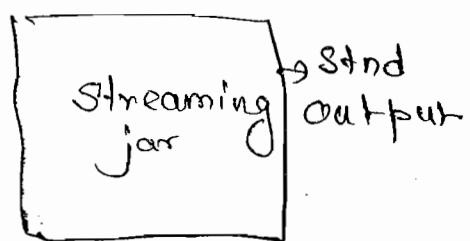
MirabI

Place single file in Hadoop Cluster \Rightarrow file is Input
 \Rightarrow copy the data

\$ bin/hadoop fs -put paste /user/vm4learning/
Weatherfile.txt

\$ bin/hadoop jar contrib/streaming/hadoop-
streaming-lzo.jar -file /home/vm4learning/
code/streaming/max-temperature-map.rb

cont. standard



→ It is a waste of time to put from one block to another therefore we use -file
cont.

-file /home/vm4learning/code/streaming/

max-temperature-reduce.rb -mapper max-temperature-
rb -reducer max-temperature-reduce.rb

-input /user/vm4learning/input4/sample.txt
-output /user/vm4learning Max-emph

Home

↳ code

↳ streaming

↳ mapper

~~F~~

FILE FORMATE

- improve your job
- default → text file formate

file formate

① TEXT →

- ↓
- can be compressed
- but cannot be splittable

PIP keys contains the boundary of each line.
and it ~~is~~ text file does not have —, —.
Therefore it is not splittable.

- If you want to be improve the performance of job, it effects the file formate.

Disadvantage →

- The txt file formate cannot be splittable after compression.
- We need compression to improve the transfer speed along ^{with} saving space.

② Sequential file formate —

↓
Key value → pairs

- which contains data in key value pairs.
- can be compressed & splittable also.

Codecs

↓ used for

compression & decompression

types of compression

- LZO
- Snappy
- Bzip2
- Gzip2

Open the material
called MapReduce

where you compress the data

Codecs

LZO
Snappy

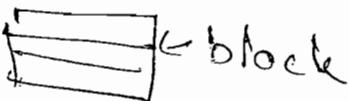
- MAP O/P
- SIP
- OIP

→ we can going to compress the data in 3 'levels'

① → File level



② → Block level



③ → Record level



→ By default compression

→ disable in Hadoop

Step ① → Enable the compression

② → Codec ← method

③ → Level ← what type of level?

Text
File → job.setInputFormat → Sequential o/p
Text file format
convert

Open
MapReduce file format.

Open Eclipse

Open Textmap.txt → inputs → MR → TextToMap.txt
d001 Marketing right click → Blank

- 1 Marketing
- 2 Finance
- 3 Human Resources
- 4 Production
- 5 Development
- 6 Quality Mgmt Management
- 7 Sales
- 8 Research
- 9 Customer Service

→ Go to properties → copy file
Convert this file into sequential file

\$ bin/hadoop fs -put /user/vm4learning/format* /user/vm4learning/format

remove that file

\$ bin/hadoop fs -rm /user/vm4learning/format*

→ /TextToMap.txt

Open →

formatConverter Text To SequenceDriver.jar
/home/vm4learning/Desktop/wordCount.jar

\$ bin/hadoop jar wordcount.jar properties
part majored-

brief. Format Converter TextToSequenceDriver

/user/vm4learning/format_input/TextToMap.txt

/user/vm4learning/format_output/

\$ bin/hadoop fs -text /user/vm4learning/
format_output/part-m-00000

To view Sequential file format

Sequential. write()

— .append()

— .read()

~~14/07/2015~~
~~Tuesday~~

Text → Sequence
(k, v) → (k, v)^{Text}

Format Converter SequenceToTextDriver.java ↳

→ To convert from sequence file format to text file format

\$ bin/hadoop jar /home/vm4learning/Desktop/
WordCount.jar mapred-brief-format Converter
SequenceToTextDriver /user/vm4learning/format_-
output/part-m-00000 /user/vm4learning/
output-format

↳ Format Converter TextToBlockCompSequenceDriver.java ↳

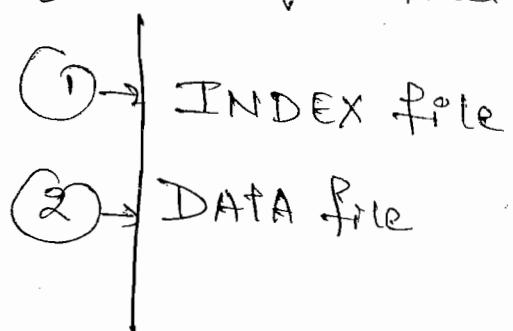
\$ bin/hadoop jar /home/vm4learning/Desktop/WordCount.
jar mapred-brief-format Converter TextToBlockComp
SequenceDriver /user/vm4learning/format-input/
TextToMap.txt /user/vm4learning/compression/

compression → Size huge → Improve transfer speed

- ① Sequential file
② ⇒ Map file :-

MAP FILE

⇒ It is a directory which consist of 2 sequential file



INDEX -

⇒ (key, value) pageno.
pairs

⇒ Random Reads

$\text{key} \rightarrow \text{key}$

⇒ contains the keys and the starting position

DATA

⇒ contain I/p data of the file

How to convert the file into map?

Mapfile.write

Mapfile.read

Information converter TextToMap.java

→ Run this file

bin/hadoop jar /home/km4learning/Desktop/
wordCount.jar mapped_brief

TextToMap.jar /user/km4learning/mapoutput
To convert binary file into text file —

\$ bin/hadoop fs -text /user/km4learning/
mapout/index

\$ bin/hadoop fs -text /user/km4learning/map
out/data

⇒ Mapfile is the random read for the sequential
combination of read & write

~~db~~

⇒ Hadoop give the file format of db Input file
format.

scr

↳ default package

↳ MyDriver.java

ctrl + m

terminal

open new tab

root | mysql

Hadoop -

mysql -u root -p

password: mmtlearning

mysql <

mysql> show databases;

mysql> create database bigdata;

mysql> show databases;

mysql> use bigdata;

mysql> show tables;

mysql> create table employees(employee_id
int primary key, name varchar(15)); <

mysql> show tables;

mysql> select * from employees;

mysql> insert into employees values(45, "Ranind");

mysql> insert into employees values(47, "pratap");

mysql> insert into employees values(48, "Raghul");

mysql> ins

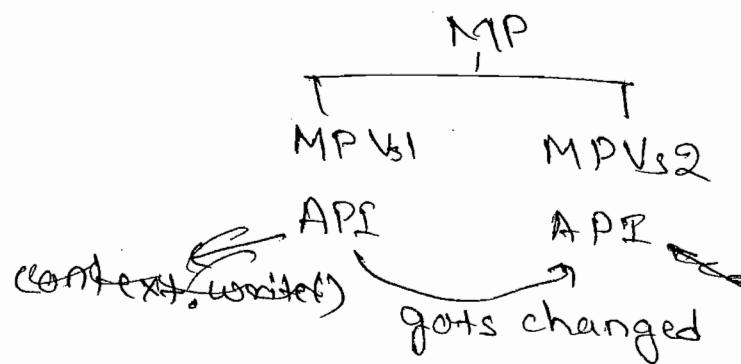
mysql> select * from employees;

Server uses → localhost
working on → BigData
username → root
password → unlearning

⇒ We do not need of "where" clause in this.

read fields } helps to write our own
write fields } data type
compareto }

My Record will going to write our own datatype.



context.write() → output.collect()

To Run new api old api
 ↑ ↑
 ⇒ Meta-INF

DBInputformat

Search → MyDriver

⇒ /home/umk-lws/learning/BigData/workspace/DBInputFormat/
Output

[Run]

Output

↳ part-00000 ↳

45	Aruntd
47	Bredep
48	Raghur

100-03 1959-12-03

44-05-01
55
59
58
59
58

employees.tsv ↳ open the file
↳ properties

\$bin/hadoop fs -put (paste) one file
vm4learning/newfile

Goto the Eclipse →

Multiple file Output format → each reducer file
for each key
→ It will create a reducer with key name
& that will be reducer come with a key
name.

Driver format Multiple Output files ↳ open
Lazy Output format ↳ Remember this

Goto

MapperFormatMultiOutput.java → open

Goto

ReducerformatMultiOutput.java → open

Now Run

```
$ bin/hadoop jar /home/vm4learning/Desktop/  
WordCount.jar Mapped-brief.&DriverFormat  
MultiOutput /user/vm4learning/mapsmultiple  
outputs/MultipleOutput.txt /user/vm4learning  
/newfile/output/
```

FILE FORMATE

5/07/2015
Wednesday

On LineInputformat

Word count

↳ src

↳ mapred (DriverNLineInputFormat.java)
— brief

/user/vm4learning/mapmultipleoutput ↳ open the file

MapperNLineInputFormat.java → open it.

⇒ Now run (DriverNLineInputFormat.java)

\$ bin/hadoop jar /home/vm4learning/Desktop/
wordCount.jar mapred_brief.DrivenNLineInput
format /user/vm4learning/mapmultipleoutputs
/multipleoutputs.txt /user/vm4learning/
NLineOutput

Now Goto Browser & check the no. of maps = 4

Q) Why Hadoop is not good for small files.

→ ~~Easy Splittable~~ → make isSplittable(false) ^{to execute sequentially}

→ Combine the 2 files → Combine file Input Format.

→ Decrease the no. of splits.

⑧ CombineInputFormat

Home/Desktop/FirstOne/Input/MR | Small files

\$ bin/hadoop fs -put
combinefile10 src /user/um4learning

Open →

~~DriverCombineInputFormat.java~~

Driver Combine File Input Format.java → Open the file

New open

⇒ ExtendedCombineFileInputFormat.java

Run the job now

\$ bin/hadoop jar /home/um4learning/Desktop/
WordCount.jar mapped_brief DriverCombinefileInput
Format /user/um4learning/combinefile/output

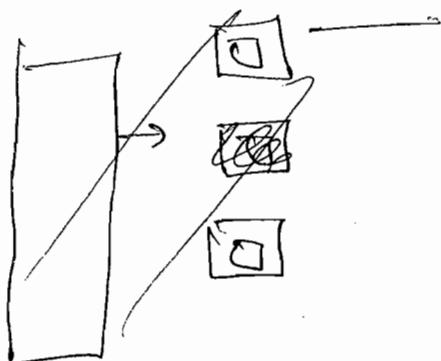
⇒ Controlling the no of mappers

using → Driver CombineInputFormat

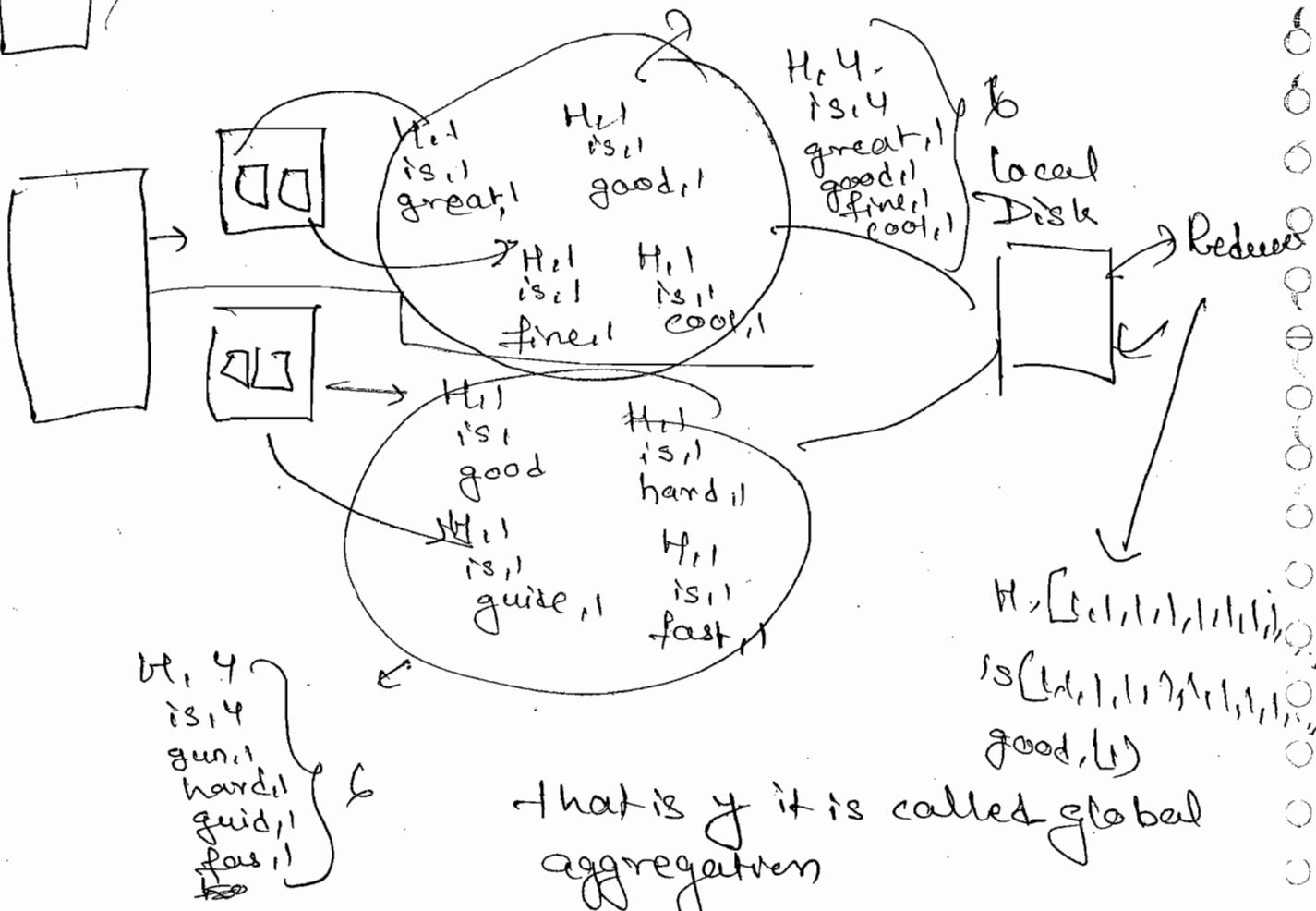
Optimize Technique -

① Combiner -

w/o combiner... what is the issues?



Here we want
to control the
no. of tuples



job.setCombiner → Reducer Code

⇒ With combiner or w/o combiner, result should be same.

- Combiners is one of the opt. tech. which will improve the performance of job.
- Combiners will do local aggregations whereas reducer will do global agg.
- Combiners are called as mini reducer, can do aggregations for data levels.
- Reducer will do agg. for across data nodes
- with / w/o combiners the result should be the same.
- If i have combiner then no need to user reduce
- If ~~one~~ data node is 1 then combiner become reducer.

Open ⇒

WordCount.java

⇒ job.setCombinerClass (IntSumReduce.class);

Input

↳ sample.txt → Copy property

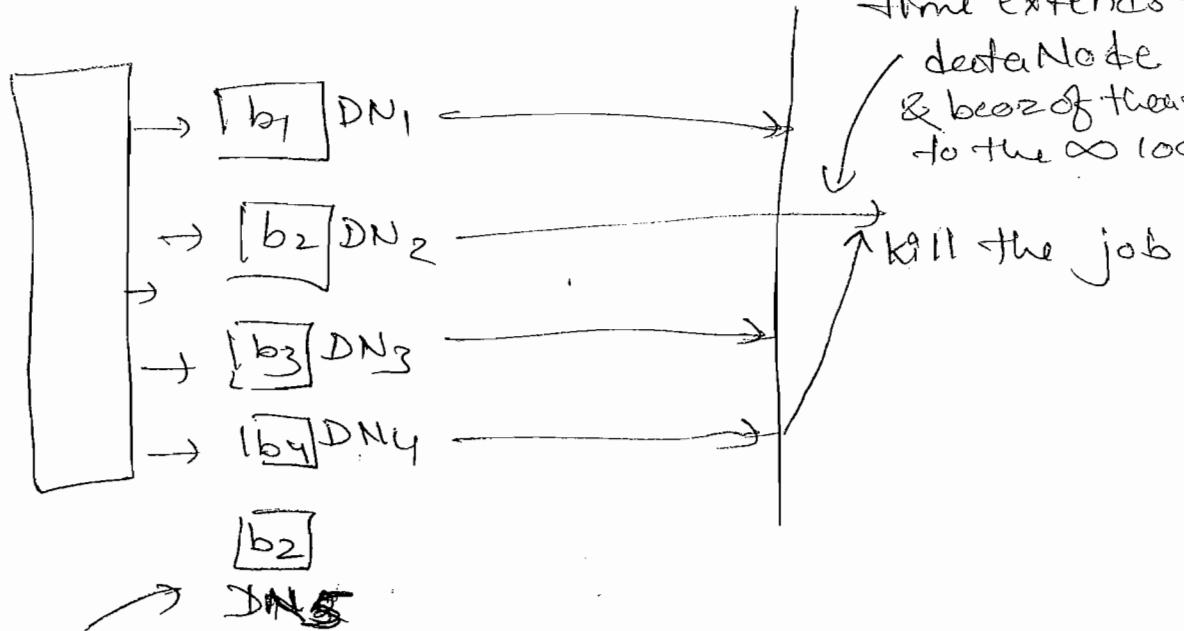
WordCount.java → Run Config.

Arg. → paste the file

↳ home/umtklearning/Workspace/BigData/WordCount/

input/sample.txt

② Speculative Execution -



b₂ goes
issue &
again
establish
a new replicated
of b1/b2 & launch
the job to replicated
to complete the execn.
The property of default will be true

⇒ And it checks whether the same job is executing
kill the job

- Speculative Execn is another opt. tech. which will ~~to same~~ launch to the same mappers launches on the replica of the block.
 - Whenever the mapper \$ go to the extended time then, At the time the mapper will go to ∞ loop.
 - Here the mapper reaches more than the avg time, in this situation, Hadoop uses speculate execution, to launch duplicate job on the job with the different data node.
 - The sp. exec. job will kill the job which takes time.
 - The default is
`mapreduce.map.speculative=true`
- Open the word count program
- => `job.setMapSpeculativeExecution(false);`,
`job.setReduceSpeculativeExecution(false);`,
- ---> make `mapreduce.map.speculative=false` to optimize the execution of job.

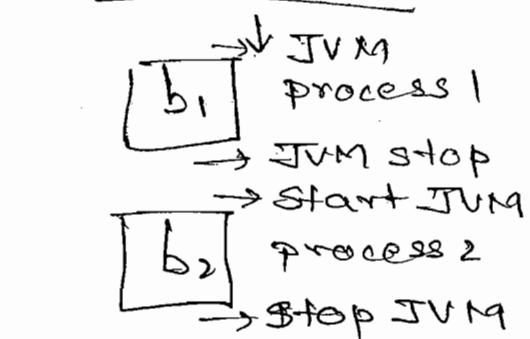
Don't hardcode it... pass the avg real time.

Open
 GraphReader
 Map
 ↳ MRLAB2.txt

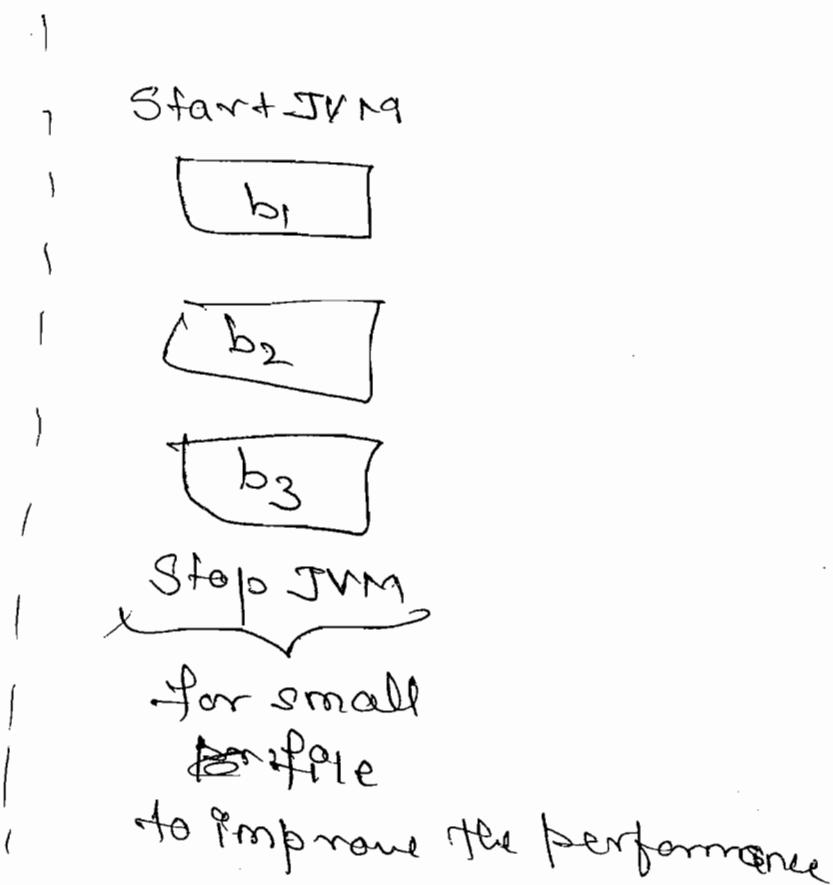
Program should implement
 tool interface to pass the
 arguments in cmd line...
 -d option

[Speculative Execution] → Open

③ JVM Reuse



Default
for 1 JVM → 1 block



- To solve small file problem/issues
- same jvm for all the block

③ Increase no. of slots, increase the performance of job.
Hm/install [hadoop12.11] config | gedit

(ii) Combiner trim

① partitioner

② Create your own file format

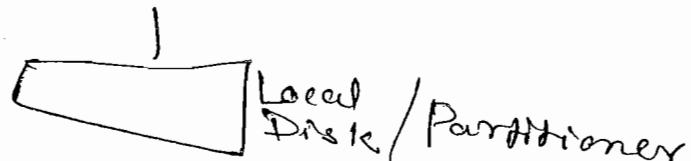
Take a big file —

① Write file format

→ which takes the 2 lines

Submit the opp in the Mapper

write the Mapper
& Combiner

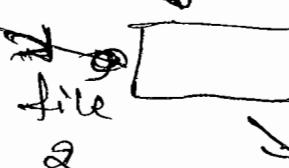


Database

get
the structured
data



Sequential
compressive
file



Convert into
text file

Requirement -

① I want to know which age group max
watches my site

ID	URL
45	www.facebook.com
46	→ / paytm → ,
47	→ / google → , —

AS url + count ← How many
times it

watches
the website
or repetition
of website

②

Create a table		
ID	age	Name

getting
in
Hadoop
files

~~16/07/2015~~
~~Thursday~~

→ Download Twitter Data

FIND Out → How many times a word (combine) repeated?

→ This is called Word Co-occurrence...

What is N-gram (Text Analytics)

Bi-gram

Tri-gram

job.setreducer
(intSumReducer.class)

← This class sum up all the words.

```
prev=null
for(String s=words)
{
    if (prev!=null)
    {
        word.set(prev+' '+s)
        context.write(word,1)
    }
    prev=s;
}
```

Assignment -

1. Word Count
2. Word Co-occurrence
3. Trending in Twitter Website

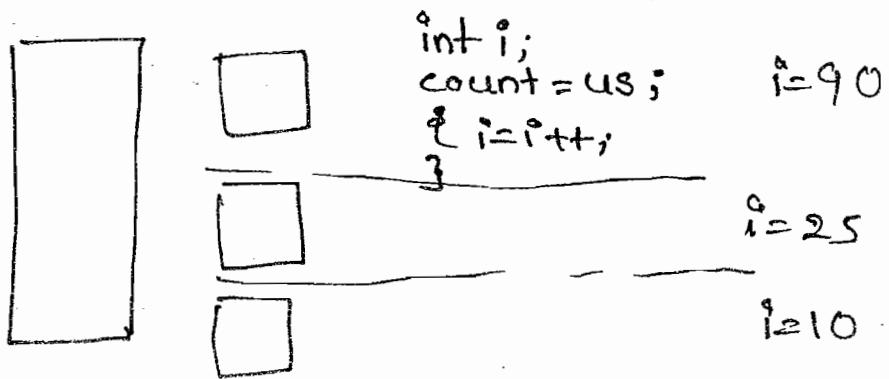
SORTING

Hadoop will do sorting of keys only not on values.

COUNTERS

W/O Counters → What is the problem?

Telephone
Directory having US in capital
and not recognise us in small letters

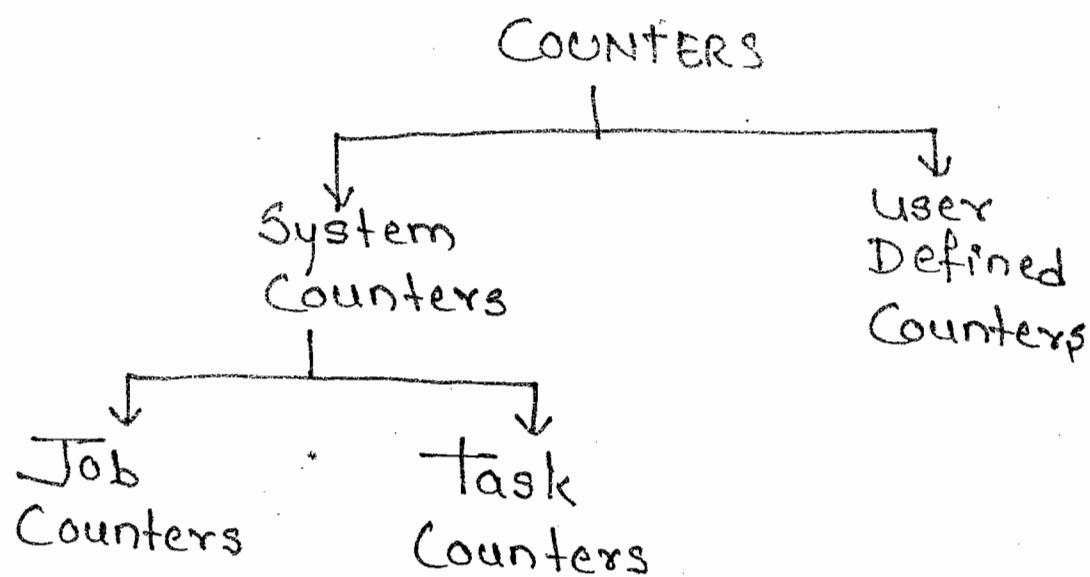


To solve the problem, we need shared variable call

→ Counters are mainly used for matrices & statistics of the job.

→ Matrices & statistics of job will help us to know how much time it will took to process the job and how many I/O operations are performed on the job.

→ Counters are of mainly two types -



→ The Counters which are emitting from task is called Task Counters.

→ The Counters which are emitting from job is called Job Counters.

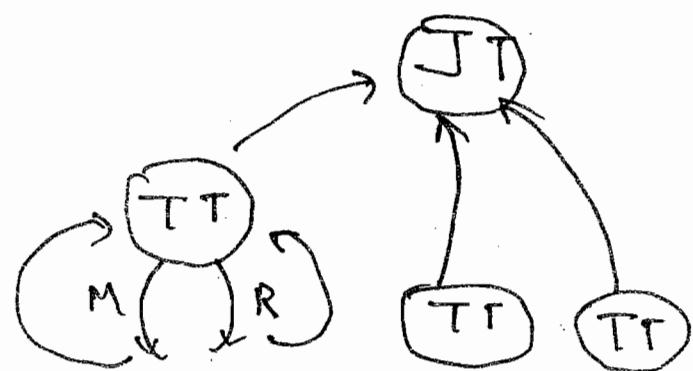


Fig:- Job Counter & task Counter

Now Start the Cluster → \$bin/start-all.sh

```
$bin/hadoop jar /home/vm4learning/Desktop/  
WordCount.jar org.apache.hadoop.examples.  
WordCount /user/vm4learning/input3  
/user/vm4learning/output
```

```
$bin/hadoop jar /home/vm4learning/Desktop/  
WordCount.jar org.apache.hadoop.examples.  
WordCount /user/vm4learning/format_input  
/user/vm4learning/output
```

Now → Open Eclipse

↳ Counter
↳ src
↳ WordCountWithCounter.java

Problem Solved -

$$\begin{array}{c} \boxed{} \rightarrow 2 \\ \hline \\ \boxed{} \rightarrow 3 \\ \hline \\ \boxed{} \rightarrow 4 \end{array}$$

Counters counters = job.getCounter() → all the system and user counters will give the enum be defined...

→ Now, Run the program in Stand Alone Mode.

O/P Number of input records = 5

```
$ bin/hadoop jar /home/vm4learning/Desktop/Counters.jar org.apache.examples.WordCountWithCounters /user/vm4learning/inputs /user/vm4learning/output
```

↑
Now, Run the program in

Pseudo Distributed Mode

→ Goto the MapReduce

↳ Number-of-Records

JOINS —

→ We can do joins in JAVA too.

→ Joins is more costly in SQL.

emp
id, name, Deptid

dept
Deptid, name

→ Actual joins happens in Reducer ~~map~~ side.

↳ It is too time taking.

Therefore, we go with another technique.

Select id, name, Deptid, name

from emp Joins dept

ON emp.Deptid = dept.Deptid

→ To perform join at map side

one of the table should be small enough
to fit into the memory.

→ Place the joins table at Distributed cache.
So that, all the DN can access that data.

→ job works parallelly but they can not communicate.

Only thing is context object by which the jobs communicated. → You can say it as a Counters

Put

① employee.tsv
② departments.txt] Put this 2 files into HADOOP CLUSTER

Which one is the small file?

→ departments.txt

WordCount

↳ src

↳ mapred_brief

↳ DriverMapsideJoinCacheTxtFile.java

```
$ bin/hadoop fs -put /home/vm4learning/Desktop/input/ /user/vm4learning/DC/input
```

Open the File/Program -

DriverMapSideJoinDCacheTxtFile.java

Distributed Cache.addCacheFile

You are simply adding file to the distributed cache

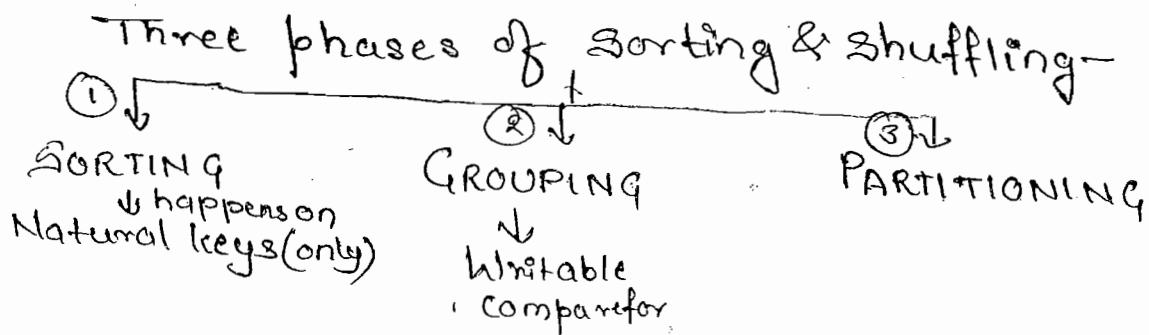
Open the Program -

DriverMapSideJoinDCacheTxtFile.java

Now Run the program

```
$bin/hadoop jar /home/vm4learning/Desktop/  
WordCount.jar mapred_brief.DriverMapSide  
JoinDCacheTxtfile /user/vm4learning/mapsidejoin  
/employee.tsv /user/vm4learning/output
```

17/2015
FRIDAY



Writable Comparable (key)
> Writables (values)

VIMP Concepts
Cover Hereoos

\$ sudo easy_install pip → install it

Now we come to Secondary Sorting -

\$ cd \$HADOOP_HOME

Eclipse
↳ Project Explorer
↳ src
↳ SecondarySortBasicDriver.java

Open Input.txt

\$ bin/hadoop fs -put /home/um4/learning/workspace
BigData/wordCount/input/Input.txt /user/um4/learning/
Sorting/output/input.txt

CompositeKeyWritable.java → we are writing our
own datatype.

Open →

SecondarySortBasicMapper.java

Open →

SecondarySortBasicPartitioner.java →

we can make
our own partition

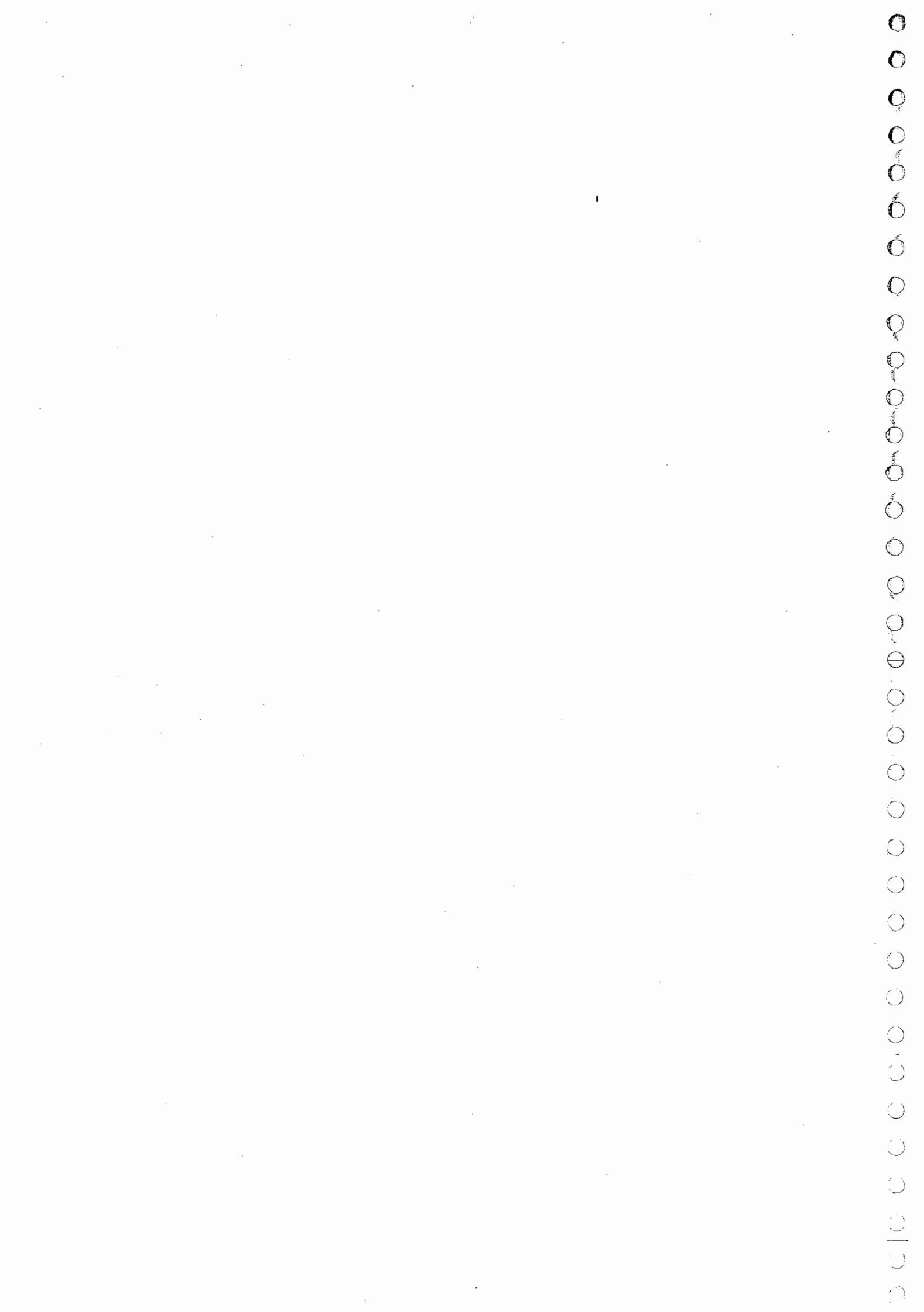
Sort Comparator - compare the Grouping

Open → SecondarySortBasicComplexKeySortComparator.java

src

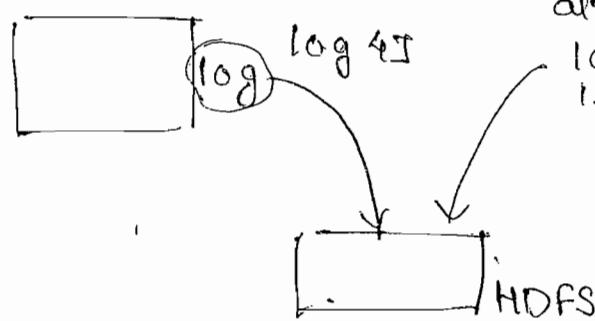
\$ bin/hadoop fs -put Input.txt /user/vm4learning/
Sorting/Input.txt ⇒ see previous cmd

\$ bin/hadoop jar /home/vm4learning/Desktop/
wordcount.jar mated_brief.SecondarySortBasic
Driver /user/vm4learning/Sorting/Input.txt
/user/vm4learning/Secsortng/



18/07/2014
SATURDAY

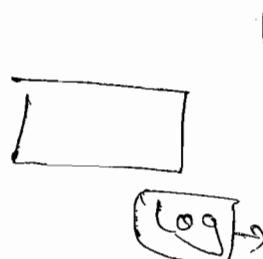
1. LOGIC



already have
100 files the how will be
ISO files mean so more

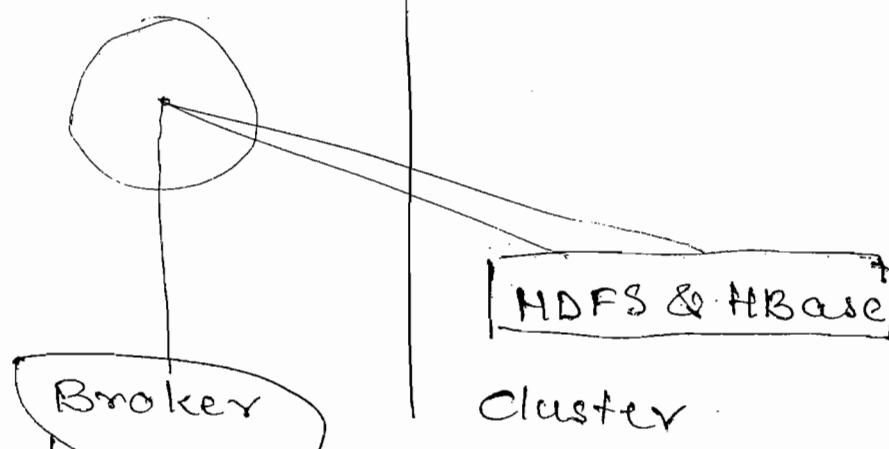
- We need a broker here
- We want directly all file into hadoop cluster
- Donot need to use put
- This are called Messaging brokers.
- Public subscribe Messaging tool... eg Scribe

(Facebook uses)



RUMI
3 main component or Agent
① Source → the one who generates the msgs
② sink → the channel
③ Sink → the one who stores the data

Application
Server



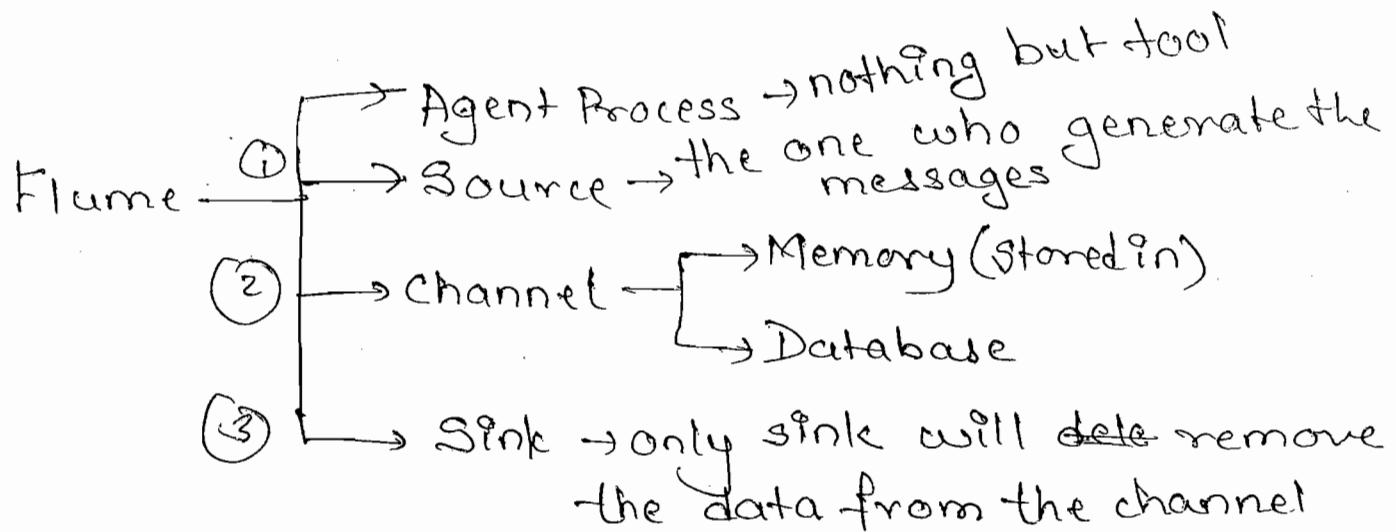
Broker
Decoupling
Source & Sink

- The place where the data is store is sink

Broker Decouples the Source (Appn Server) & sink (HDFS OR HBASE)

Flume is log aggregation tool.

↓ it is the best tool
↓ integrated with
HADOOP



- channel data will be very fault-tolerance
- therefore, it stores the data in database
- Memory (channel) can be delete the stored data therefore, we use database.

Kafka → is used to ^{process} ~~same~~ 1 billion records in 15 minutes.

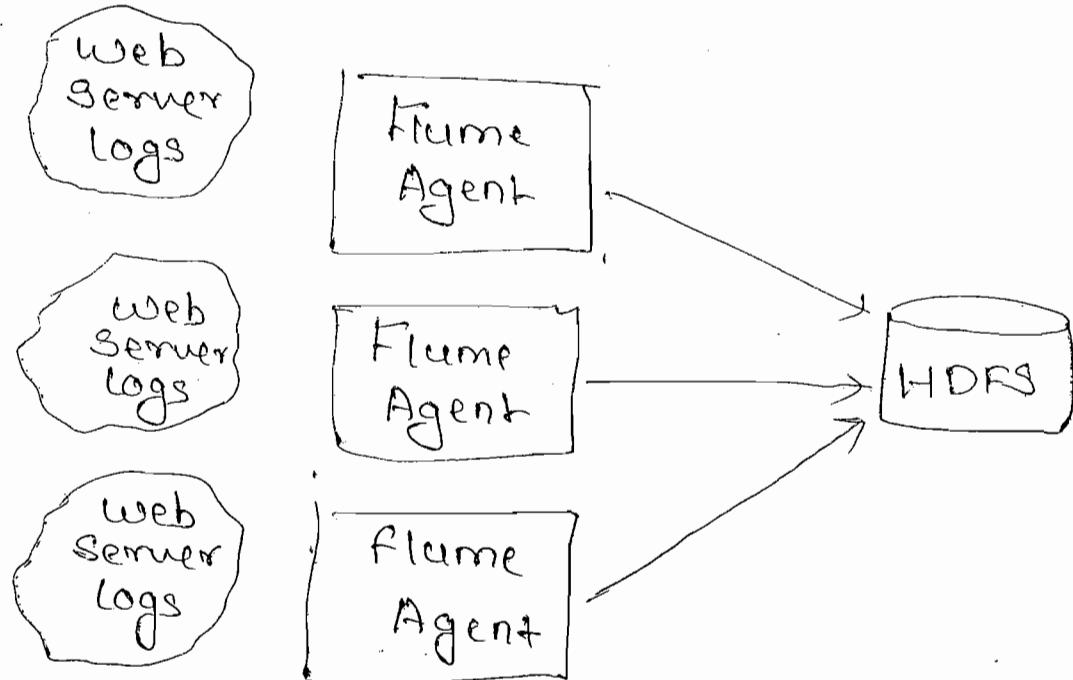
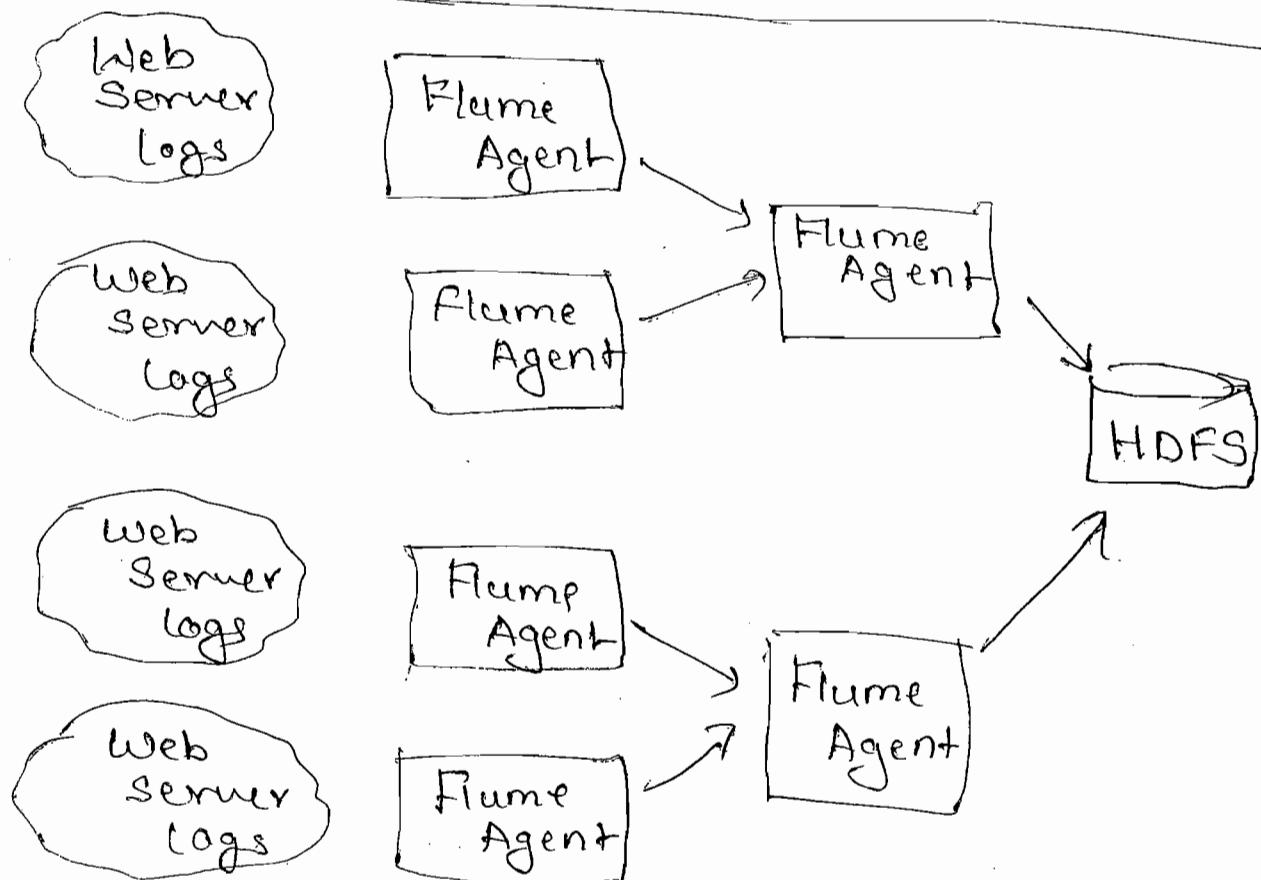


Fig: Single Agent Topology

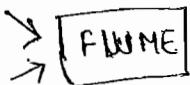
One source → One Agent → HDFS



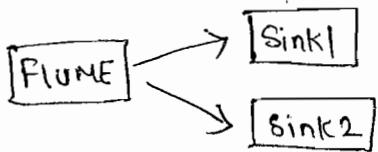
Multi Agent Topology

There are two types of operations -

① → FAN-IN



② → FAN-OUT



Agent ...

Start → Hadoop cluster

\$ bin/start-all.sh

\$ cd \$FLUME_HOME

\$ ls

\$ vi srujan.txt → virtual editor

\$ This is Monday

This is Tuesday

—, —

escape :wq

\$ tail -F srujan.txt

This is Monday

This is Tuesday

This is Wednesday

This is Thursday

This is Friday

This is Sat

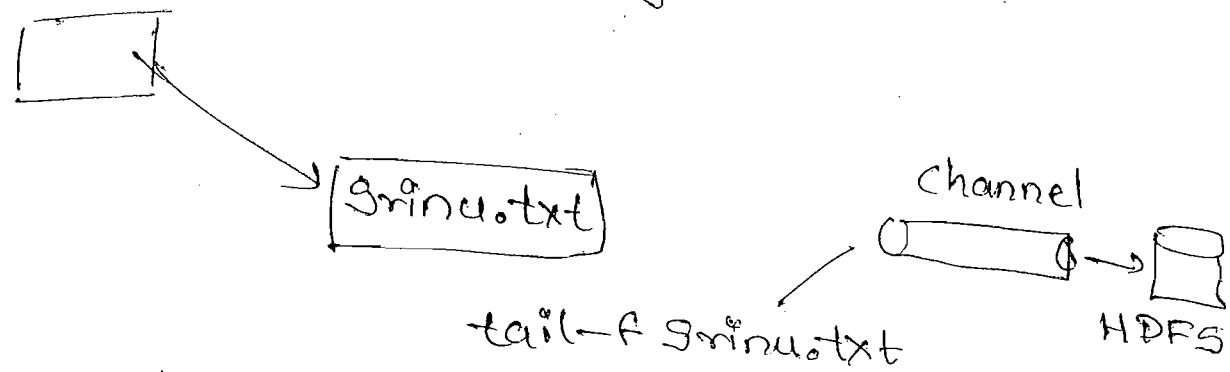
This is Sun

escape :wq

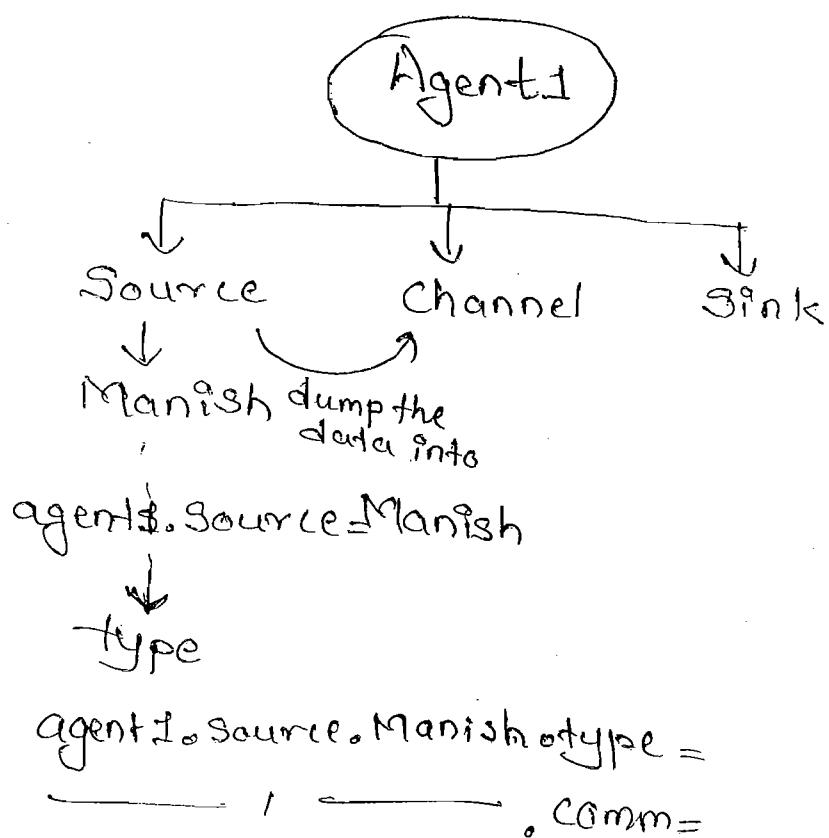
} \$ vi srujan.txt

\$ tail -F Srinu.txt

→ w/o opening or running Srinu.txt,
the tail cmd automatically fetches data
updated w/o running again the cmd.



Open Configuration file
↳ flume-agent.conf



Go to

↳ Flume.txt ↳ home | installations | apache-flume-

\$ bin bin

\$ ls

\$ bin / flume-ng agent

flume.txt
bin

O → is used to

get u in
next line

agent -

— conf o / conf /

-f o / conf / flume-agent / .conf

agent1

\$ rm srujan.txt

\$ bin / flume-ng

Now, we will see HBase as a sink.

flume-hbase.conf

Open in a new tab
Hbase

\$ cd \$HBASE_HOME

\$ bin / start-hbase.sh

\$ bin / hbase shell

001> list

002> create 'mytable', 'cf1'

003> scan 'mytable'

\$ Open

Log4jExample.java

→ Log4jExample.java

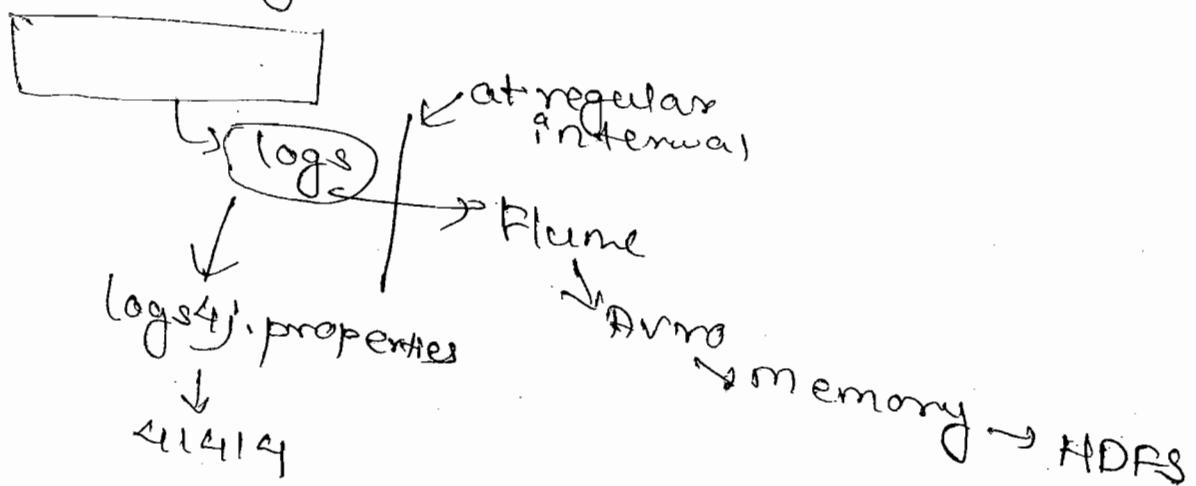
Log4jProperties.java

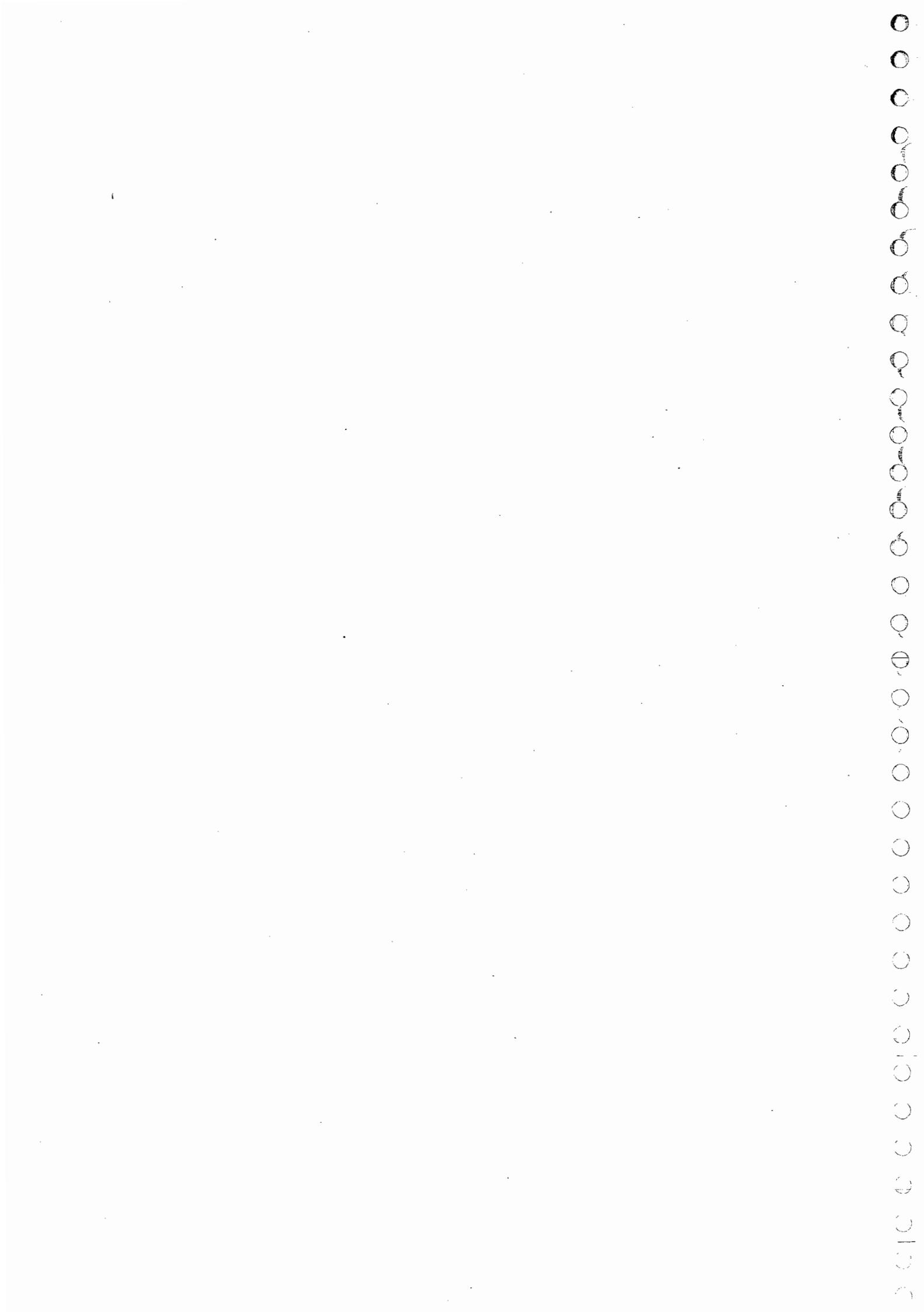
Avro → is a structured binary file

Injection tool

How data injected to Hadoop, we will have Flume tool.

Hospital mgmt

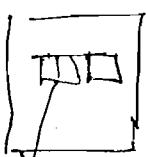
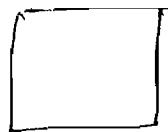




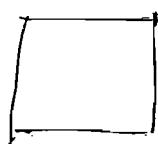
19/07/2015
SUNDAY

\$ cd \$HBASE_HOME

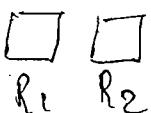
\$ bin/start-hbase.sh ← Start the HBase



Region
Servers



← Split it into



↑
Region

→ we are not distributing
the RS

HOTSPOT → ?

When you want to divide the table, you can divide it in (Row Keys).

Row Keys



0 - 100 → R1
100 - 200 → R2
200 - 300 → R3 } Splits

Lets see → How to do the splits on table

Creating a table with splits

\$ bin/hbase shell

Browser

↳ hbase

↳ hivembase → 1 split is there only (at present)

001:0> create 'mytablewithsplits', 'cf1', {SPLITS=>
['111111', '222222', '333333', '444444'] }

↑
it splits
into 5
(5 splits)

002:0> describe 'mytablewithsplits'

003:0> scan 'mytablewithsplits'

Now Goto HBase Master → Browser

↳ Goto RegionServer

↳ We can see that splits is created

004:0> put 'mytablewithsplits', '333337',
'cf1:name', 'Grinu' ⇒ Distributing
the data

005:0> scan 'mytablewithsplits'

Now we going to split distributed the data
in different regions

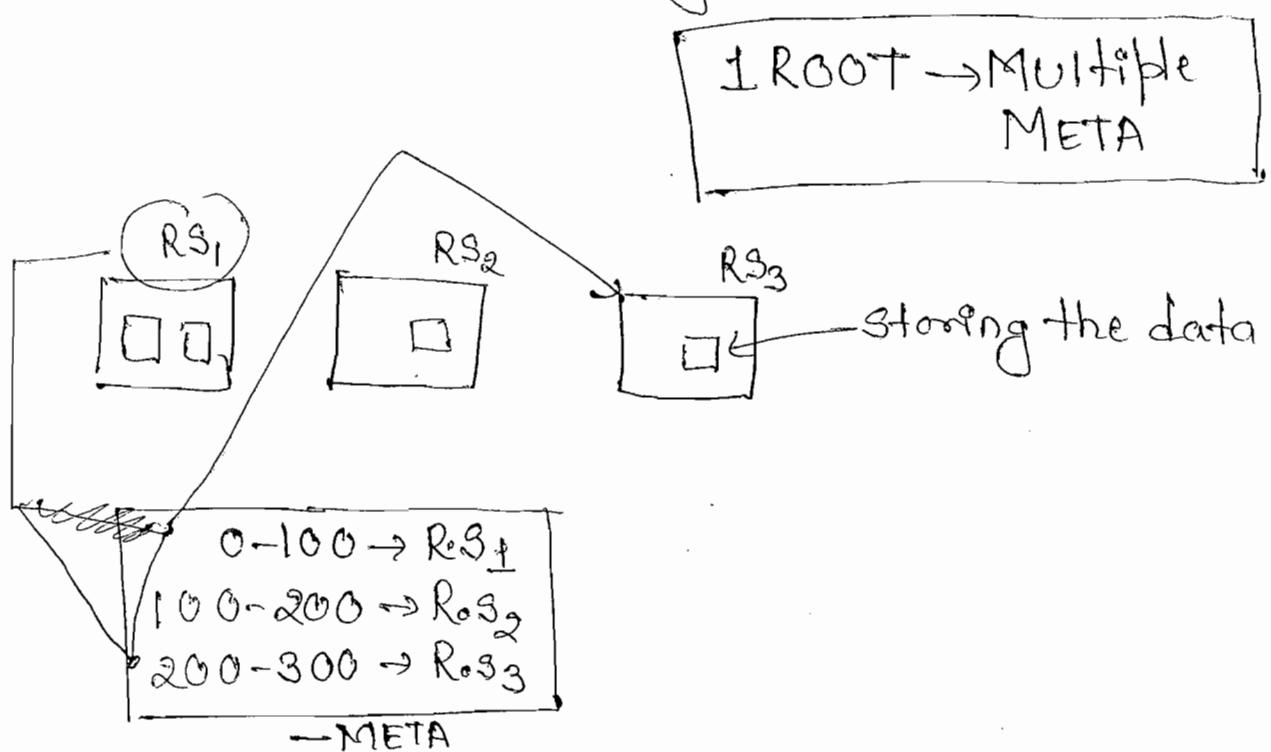
\$ bin/stop-hbase.sh

Goto → /hbase/mytablewithsplits

In the same way, you can distributed the
data in all the region server.

WHO will drive your Hbase Master?

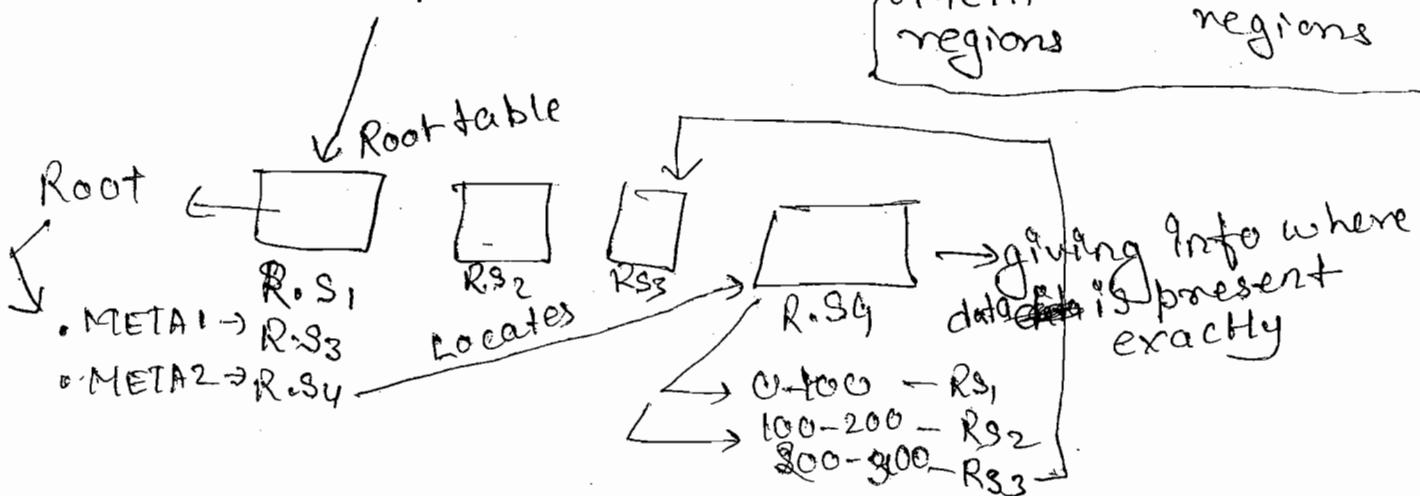
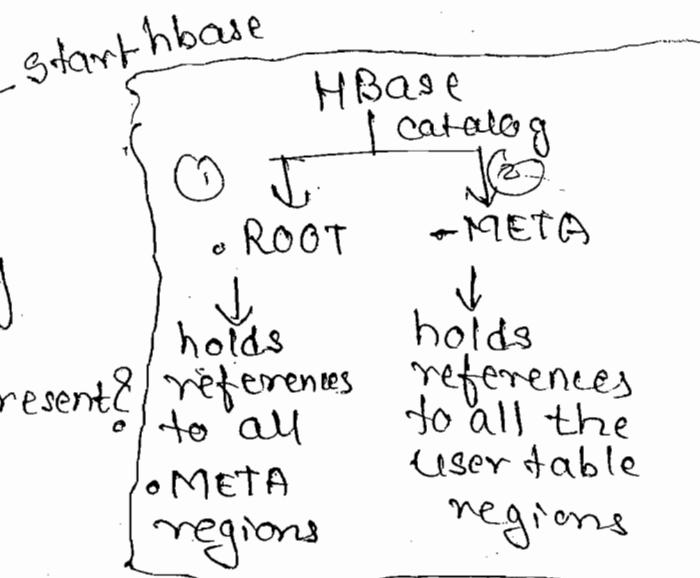
Where we store data → in RegionStore

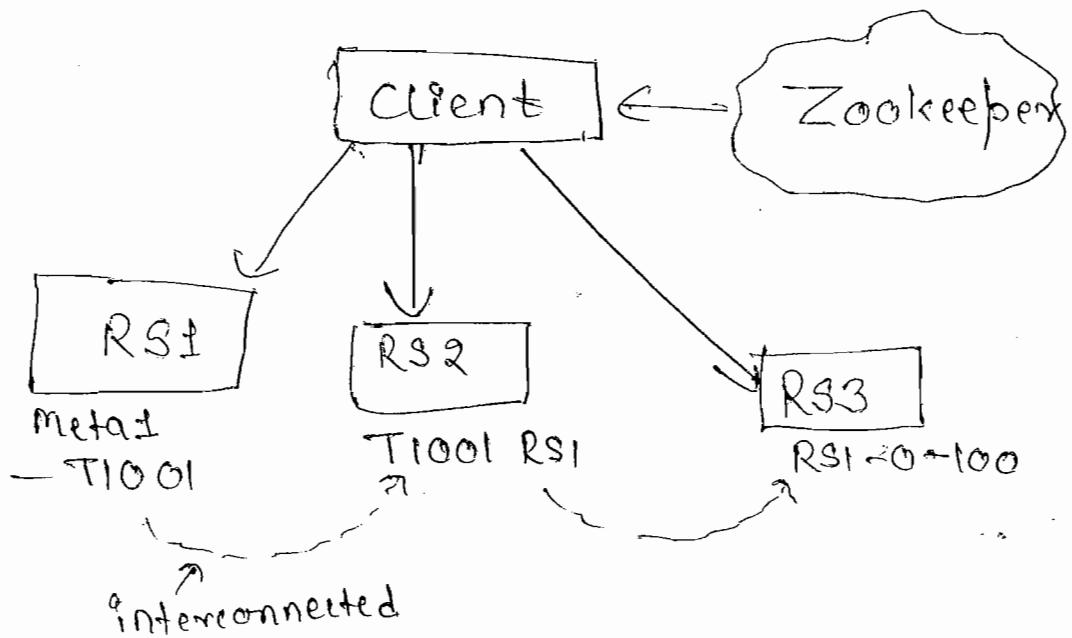


\$bin/start-hbase.sh

→ Each table will be having
1 meta

request come → ZK → RS₁
where the root is present?





Geto program →

FreqCounter.java →

Counters_Example.java →

\$bin|stop-hbase.sh
\$bin|stop-all.sh

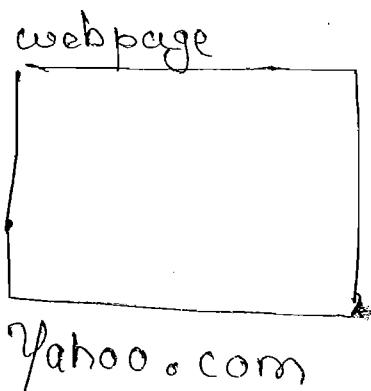
Open ⇒

Installation → hadoop-1.2.1 → conf → hadoop-env.sh
exportHadoop-Classpath ← uncomment
this codeline

Now

↳ start Hadoop → \$bin|start-all.sh
↳ start HBase → \$bin|start-hbase.sh

Assume I have
10 Users



Which user browsed the website more?

Open → Counters.Example.java → Here we have Access-logs

002:0>list

003:0>scan 'access-logs'

No record

004:0>

Client Side Buffering -

when you

enter

the data

store

in

log

and

you

can

access

the data

fast

It can be used to for fast processing.

If the spilling(freshing) stop, data will be lost, if it will stop, there can be case of data loss.

Let the data to be loss we can have the data in our logs(we can get).

write

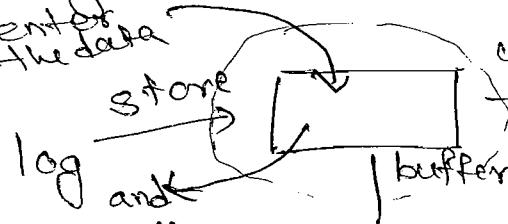
main store

Stop spilling

it will be spilling 1000 record for a 1 hbase

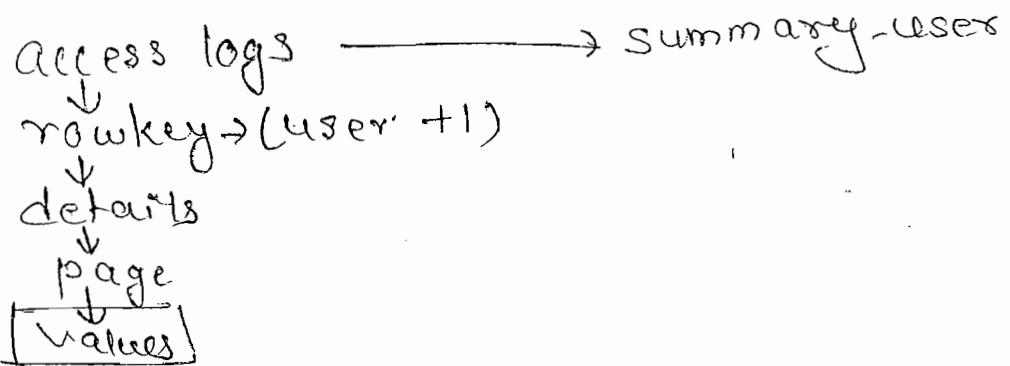
therefore stop spilling

Hfile1, Hfile2, Hfile3



- → It can be used to for fast processing.
- → If the spilling(freshing) stop, data will be lost, if it will stop, there can be case of data loss.
- → Let the data to be loss we can have the data in our logs(we can get).

For every record we want unique key.



htable.flushCommits(); → to flush the data put into the log file → goto to flume → assignment.

Run the programme

↳ Counters-Example.java → O/P →
importing 10,000 records
Done

005:0> Count 'access-logs'

006:0> scan 'access-logs' → 10,000 records are scanning with in 28.8370 seconds

007:0> create 'summary-user',

{NAME => 'details', VERSIONS => 1}

008:0> scan 'summary-user'

Open

↳ FreqCounter

↳ FreqCounter.java → Explained

JobTracker → HMaster → HDFS → now

JobTracker → Namenode → earlier

TableMapperUtil.^{init} & TableMapperJob

Table Name → Access logs

HBase-CRUD

↳ export as jar file

```
$ bin/hadoop jar /home/um4learning/Desktop/  
HBASE-CRUD.jar hbase_mapred.t.FreqCounter.jar
```

010:0> Scan 'summary-user'
10 rows in 0.0160 seconds

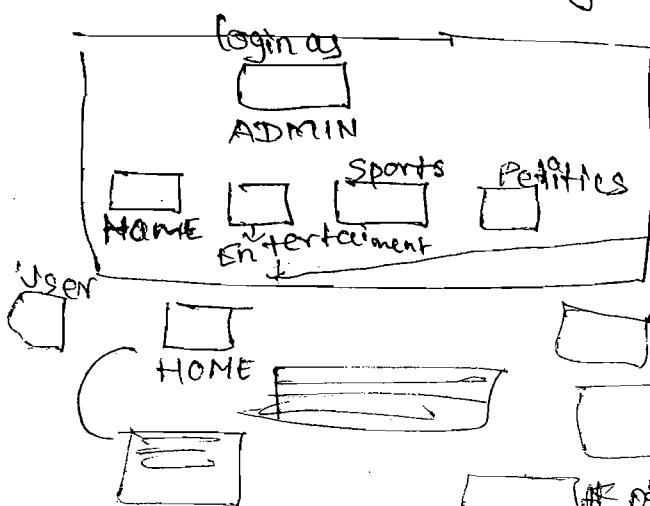
Open

↳ PrintUserCount.java → RUN

```
$ bin/hadoop jar /home/um4learning/Desktop/  
HBASE-CRUD.jar hbase_mapred.t.PrintUserCount
```

Assignment -

Create a web page



Create a ~~table~~ data access logs into the table

1. politics

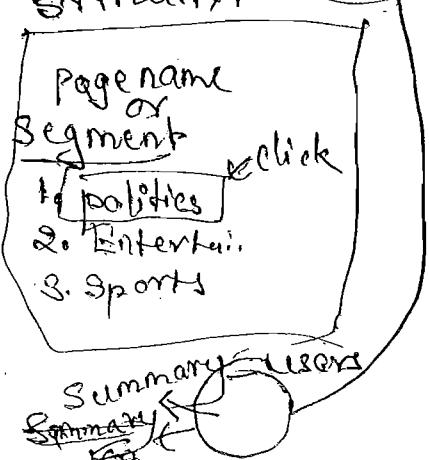
File: sminu.txt
Format: sminu
Access log segment

loginid : UserId

Politics Table

Sports Table

Entertainment Table



Summar-user

watch

- 1, politics, 980 time
- 1, sports, 3
- 1, Entertainment, 1

Politics sports

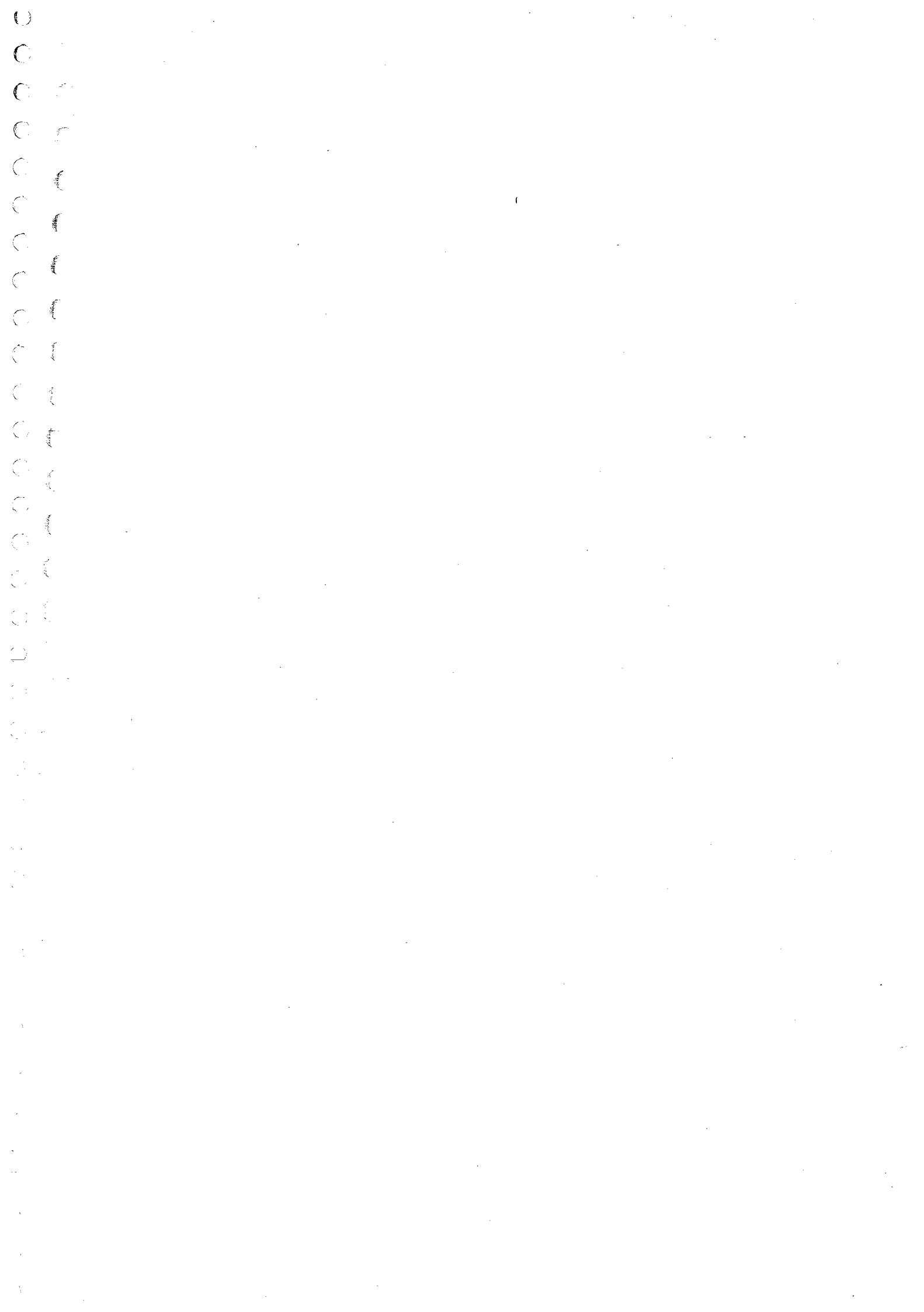
1 [980, 3, 1] → By the Reducer method

↓
enter

By the position, we can recognize
what is most watched things...

[↓ → politics]

So, in home page
we want to display only politics news
from twitter...



HIVE \Rightarrow

20/07/2015
MONDAY

FRAMEWORK — HIVE

Facebook ← HIVE

HR

HDFS | HBASE

→ To write the job in terms of SQL.

→ It is a datawarehouse tool.

→ Hive use HQL → Hibernate Query language
↓ exactly different

Hive Query language

HQL
→ which is very similar to SQL.

→ we don't want to install in Hadoop cluster
we install it in Hive...

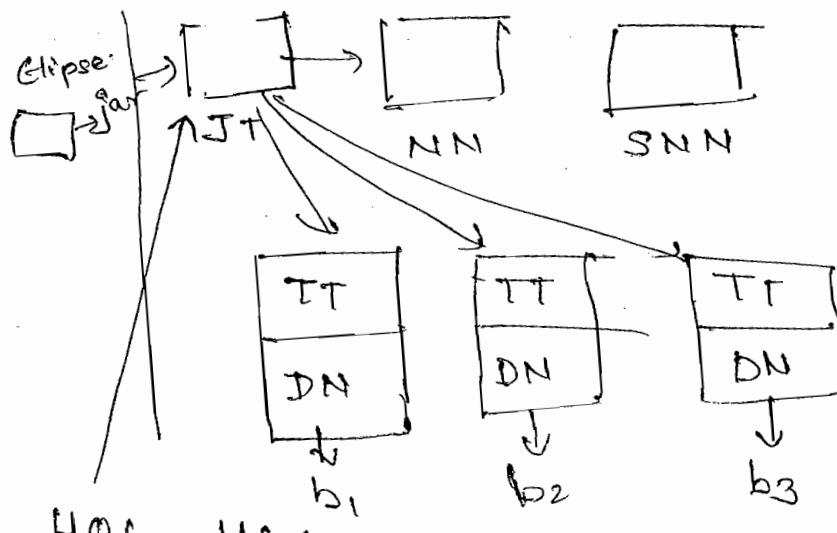
SQL $\xrightarrow{\text{HQL}}$ Java

→ Hive is only for structure data.

→ It is easy to write job in terms of MapReduce.

→ Hive is a client tool.

→ Hive is a processing tool not a database.



HQL — Hive — M-R (mapReduce)

client tool

- To write the programs in ellipse is very costly
- It is time consuming
- To have a little change in code takes lots of time
- for this reason we have Hive

HQL → Hive → M.R.

Whatever we did earlier in Hadoop cluster
Now we can do it by Hive

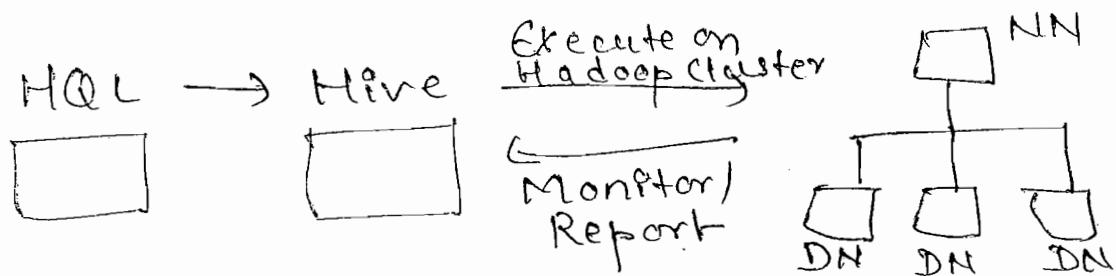


Fig. → How Hive will work?

Steps - For Running with the Cluster -

- ① → Start -allsh
- ② → place a file into HDFS

Go to

Home/Code/Hadoop-book-3e/input/ncdc/microtab
/sample.txt

↳ select the location (copy)

\$ bin/hadoop fs -put paste(snc) /user/vm4learning
-ng/sample.txt

In RDBMS

Schema + data

→ [Derby - RDBMS]

Metastore

default

→ Hive uses db called Derby Database.

→ The db of Hive called as Metastore.

Right now we don't call it as db, we call it as a Metastore

③ → Start the Metastore -

Make a new tab called HIVE-METASTORE] 2 tab
—————>—————> HADOOP

\$ cd \$HIVE_HOME

\$ pwd

Goto Installation

↳ Hive-0.11.0 bin

↳ conf

↳ hive-site.xml → open that

↳ hadoop-default.xml.template → open that

↳ you can have all the default db/metastore

Hive-Metastore

\$ bin/hive --service metastore &

→ Home/installation/hadoop-1.2.1/conf/kosherhadoop-env.sh

→ first you have to

comment the only

the cmd will run successfully...
successfullly...o.o

↳ comment the line

export Hadoop
Classpath

Open another tab → HIVE-SHELL

② bin/hive

\$ bin/hive ↪ log you in Hive-Shell

hive>show tables → it will show you some tables
hive> and the table names we got
from the ~~derby db~~
default

⑤ Create a table for the file and ②nd step

Go to browser

↳ /user/vm4learning)

1950	0	1
-1-	22	1
-1-	11	1
1949	11	1
1949	70	1

↓ ↓ ↓
Year temp quantity

you can see this type of data into it

- ⇒ ① Records are separated by tabs
- ⇒ ② Format of file is Row type
- ⇒ ③ Structure

→ (year int)
 (temp string)
 (quantity int)

hive> CREATE TABLE records (year STRING,
temperature INT, quality INT) ROW FORMAT
DELIMITED FIELDS TERMINATED BY '\t';

hive> show tables;

hive> describe records;

Goto

Hive

↳ de hive-default.xml

hive.metastore.warehouse.dir → where the
table folder
should be created
in db

browser

↳ /user/hive/warehouse/records

hive> drop table records;

browser

↳ /user/hive/warehouse

Now again create the same table → records
& again check

browser

↳ /user/hive/warehouse/records

hive> select * from records;

hive> describe extended records; → very imp cmd
→ It gives detailed information of table.

③ Load ②nd file into table ⑤
the data

```
hive> LOAD DATA INPATH '/user/um4learning/  
sample.txt' INTO TABLE records;
```

→ Whenever you load the file from any location into table, the file (src) is deleted.

Goto

/home/um4learning/records/sample.txt

Goto

HDFS /user/um4learning → no records

```
hive> select * from records
```

Now, we want to find the maximum temp.

```
hive> select * from records where year = '1950';
```

→ The line is showing map method.
→ "where" is working as filter.

Open MapReduce Console -

↳ it will show you
the record query is running
therefore the SQL is converted into java
code
→ within the single code we did the job.

hive> EXPLAIN select * from records where year='1950';

Open Map Console

↳ see it is only map job... 100%
→ the hive framework will do all the job,
no need to worry about it.

hive> select max(temp)
from record group by
years;

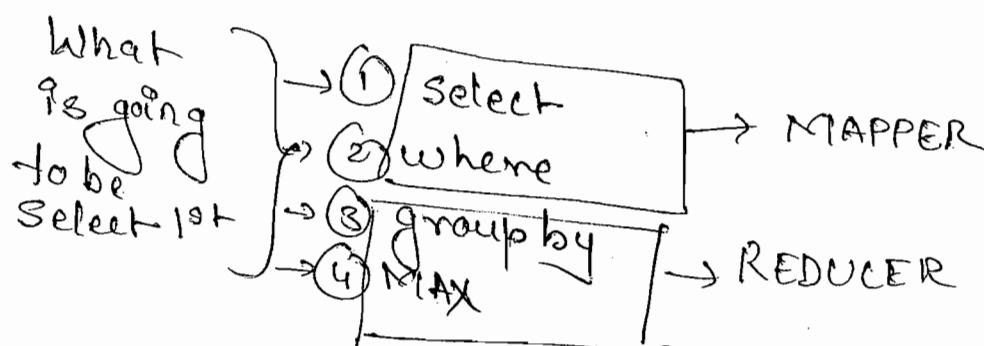
year { temp
↓ key } ↓ value

map()
{ y, +

1950, 0
1950, 22
1950, -11
1948, 111
1948, -76

1950[0, 22, -11] → Max Temp:
1949[111, -76] → 1950, 22

hive> SELECT year, MAX
(Temperature) FROM records
WHERE temperature != 9999
AND (quality=0 OR quality=1
OR quality=4 OR quality=5
OR quality=9) GROUP BY year;



⇒ that means we did MAPREDUCE job
Now, say enter the query...

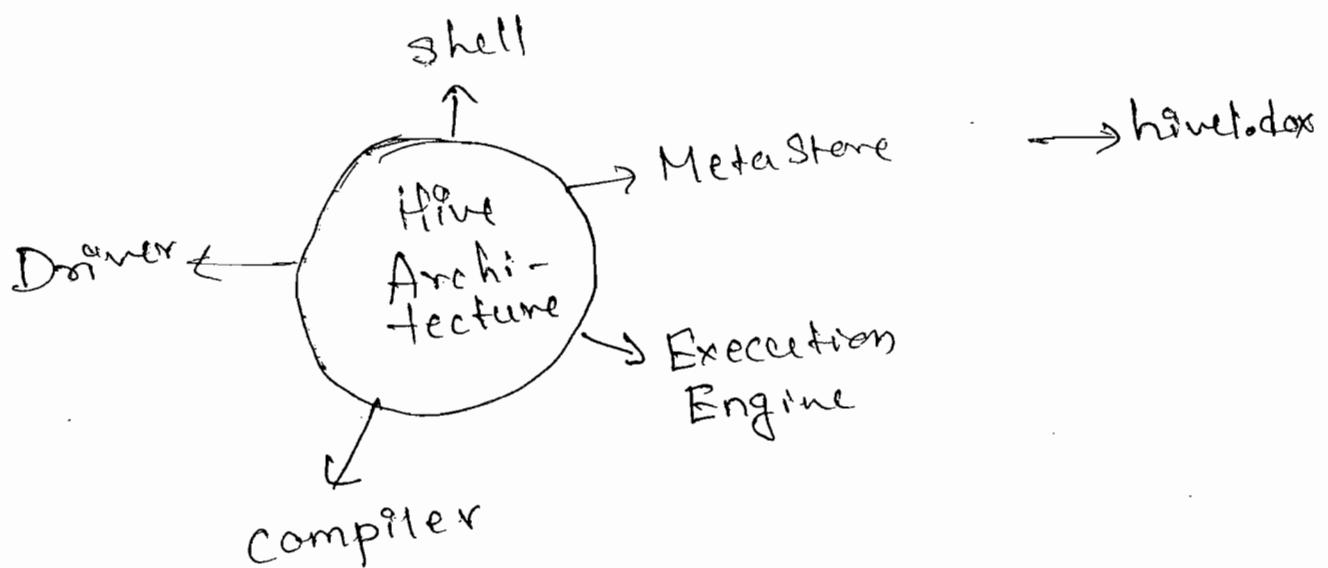
Goto MapReduce console

↳ look within 2 seconds we finish the job.

we have o/p as

1949	111
1950	22

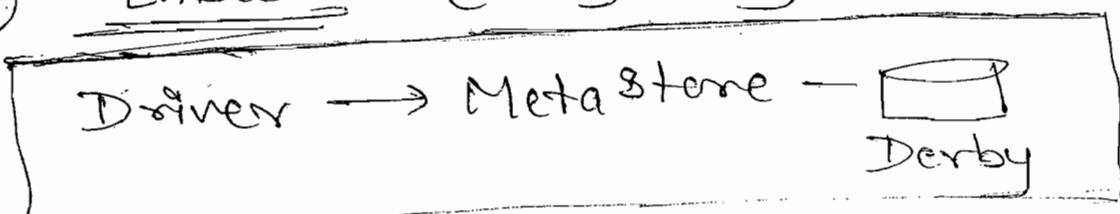
What is hive?



→ there are different types of Metastore -

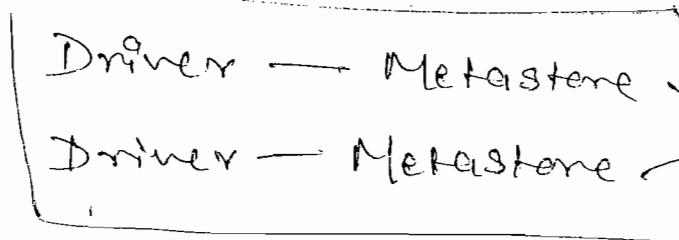
- ① Embedded Metastore
- ② Local Metastore
- ③ Remote Metastore

① Embedded - (derby → only for user)



→ It cannot be shared with others

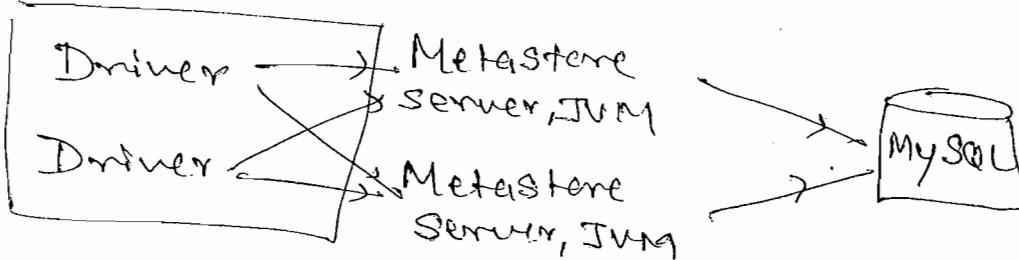
(2) Local Metastore -



→ Multiple user can access

→ But other user from another can not store
→ It is used for test tool

(3) Remote Metastore -



→ Hive uses db to store ~~data~~ schema in file.

07/20
Tuesday

Preview → What we have done in previous class → 20/07/2015 (Monday)

① Start the cluster

② HDFS

③ Metastore

④ Hive (HQL)

⑤ Create the table

⑥ Loading

→ drop the previous records (table)

\$ bin/hive

hive> CREATE TABLE records (year STRING, temperature
INT, quality INT) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\t'; ←

① HDFS → Hive table

① Create → /user/hive/warehouse/table_name
↓
hive-site.xml

hive> select * from records; (we do not get any data)

hive> LOAD DATA INPATH '/user/vm4learning/sample.txt'

OVERWRITE INTO TABLE records;

→ After deleting the data will be overwritten.

hive> select * from records; (Now we get the data)

another
data

Local → HDFS → Hive table
Mission

Local Mission → Hive table

Now, again drop the table "records"

hive> drop table records;

hive> CREATE TABLE Rrecords (year STRING, TEMper-
ature INT, quality INT) ROW FORMAT
DELIMITED FIELD TERMINATED BY '\t';

hive> Now we load the local to Hive

Local → Hive

hive> LOAD DATA LOCAL INPATH '/home/vm4lear-
ning/code/hadoop-book-3e/input/hdfs/micro-
tab/sample.txt' overwrite into table records;

Again drop the table called "records";

hive> drop table records;

hive> Now Again create the table

Hive_lab.xml

Create a table without overwrite

copy the two files < run it into the Hadoop cluster.

/Installation/hadoop-1.2.0 \$

sample.txt

sample-copy.txt

} bin|hadoop fs -put 'src' 'dst'

Now

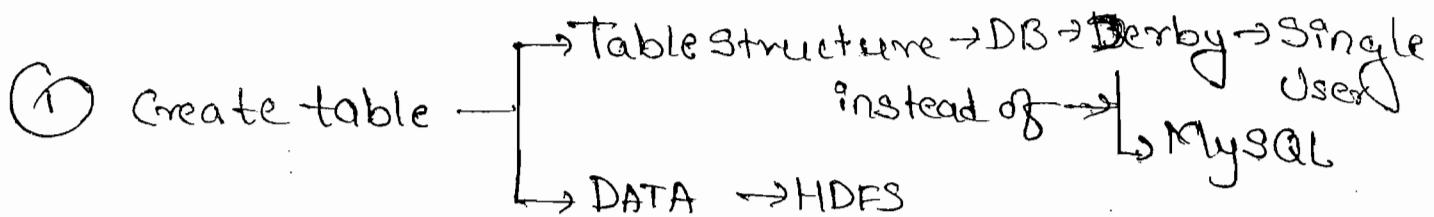
go to hive> LOAD DATA INPATH 'user/vm4learning/
sample.txt' INTO TABLE records;

hive> select * from table records;

This time we will not call for overwrite

```
hive>LOAD DATA INPATH '/user/hive/warehouse/  
/user/um4learning/samples-copy-1.txt'  
INTO TABLE records;
```

```
hive>select * from records;
```



Instead of Derby, we use MySQL

Open Another tab → MySQL

```
$ mysql
```

```
mysql> show databases;
```

Goto → Home\Installation\hive-0.11.0-bin\conf

Open → hive-default.xml.template

open → hive-site.xml

<!{property}>
comment hcatalog

<property>
→
comment

now copy from <property> to <property>
and paste it after comment.

localhost/Morning9AM?createDatabase

Come out from hive

and also stop hadoop cluster.

Now again start the cluster.

\$ cd \$HIVE_HOME

\$ bin/hive --service metastore &

hive> show tables; → it will not show any table,
because we have created
Goto my sql - new database as Morning9AM

mysql> show databases;

Now you have added

Morning9AM

mysql> use Morning9AM;

mysql> show Morning

mysql> show tables;

→ It will show all the system tables;

mysql> select * from TBLS;

→ we will not get anything

Empty sets

mysql>

hive-default.xml.template

Create the table from records;

hive> create Table records (year String, --)

Goto mysql

mysql> select * from TBLS;

→ Now you can see the data structures in the form of tables.

mysql> select * from COLUMNS_V2;

into

hive>

LET'S TALK ABOUT "DATATYPES" IN HIVE -

DATATYPES



① PRIMITIVE DATATYPES

(It can store one variable at a time)

eg-> INT, STRING

(whereas complex can store more than one variable at a time)

eg- ① Array

(is a colln of similar data types)

② STRUCT

(colln of Heterogenous DT)

③ MAP

(Collection of key-value pair) [K,V]

④ UNION → we donot use it much

Desktop | Firstone | Inputs | Employee.txt

Employee.txt

① Row FORMAT

② DELIMINATE FIELDS TERMINATED by ','

③ Create table employees

Name String,

Salary Int,

Subordinates Array[String]

Deductions ~~MAP[K,V]~~ ~~[k,v]~~ [Int]

) Collection of names separated by ','

MAP items are separated by colon

Hive-LAB -

Copy →

prashant,10000,turna Dash

and paste it into

Employee.txt

copy
this line

hive> create table employees

> address

> raw

>

> stored as textile ↴

```
hive> select * from employees;
```

```
hive> LOAD DATA LOCAL INPATH '/home/um4learning/Desktop/Firststone/inputs/Employees.txt' INTO TABLE employees;
```

```
hive> select * from employees;
```

```
hive> select name, salary from employees;
```

```
hive> describe employees;
```

```
hive> select subordinates from employees;
```

```
hive> select subordinates [0] from employees;
```

```
hive> select subordinates [1] from employees;
```

```
hive> select deductions from employees;
```

```
hive> select deductions ["PF"] from employees;
```

→ By this we can have the separate key
value pairs.

```
hive> select deductions ["IT"] from employees;
```

```
hive> hive select address from employees;
```

```
hive> select select address.street from employees;
```

```
hive> select address.street, address.city,
```

```
deduction ["IT"] from employees;
```

{ Array [MAP]
struct [MAP]
MAP [ARRAY] }

tomorrow
we will see
complex datatype on
twitter.com

22/07/2015
Wednesday

Open → Hive-LAB2

1st Start Hadoop Cluster → \$ bin/start-all.sh

Now, Start Hive → \$ cd \$HIVE_HOME

\$ bin/hive --service metastore

If you are in safe mode, you can start metastore but you cannot query.

\$ bin/hive

hive> show tables;

hive> drop table records;

hive> create Table records --

hive> load Data local \$path --

Open → Hive-LAB2.txt

copy> set hive.cli.print.current.db;

hive> set hive.cli.print.current.db=true

hive(default)> create database bighive;

hive(default)> use bighive

hive(default)> use default;

control+z ← come out of session

\$ bin/hive ← again login

\$HIVE_HOME

• HiveRC → for ever

bin/hive>

Control+z

\$ ls -a

\$ cat .hiverc

Goto Home folder → by cd

\$ cd ~

\$ pwd

\$ vi .hiverc → file in home folder
press i → press 'q'
set hive.cli.print.current.db=true;
set hive.cli.print.header=true;
press escape and ~~esc~~ say wq

Goto the Hive Home -

\$ bin/hive

hive(default)>

ctrl+z

\$ cd

\$ ls .hive* → open the hidden files

we have .hivehistory

\$ cat .hivehistory

Now we can see all the History what we
have done earlier in Hive...oo

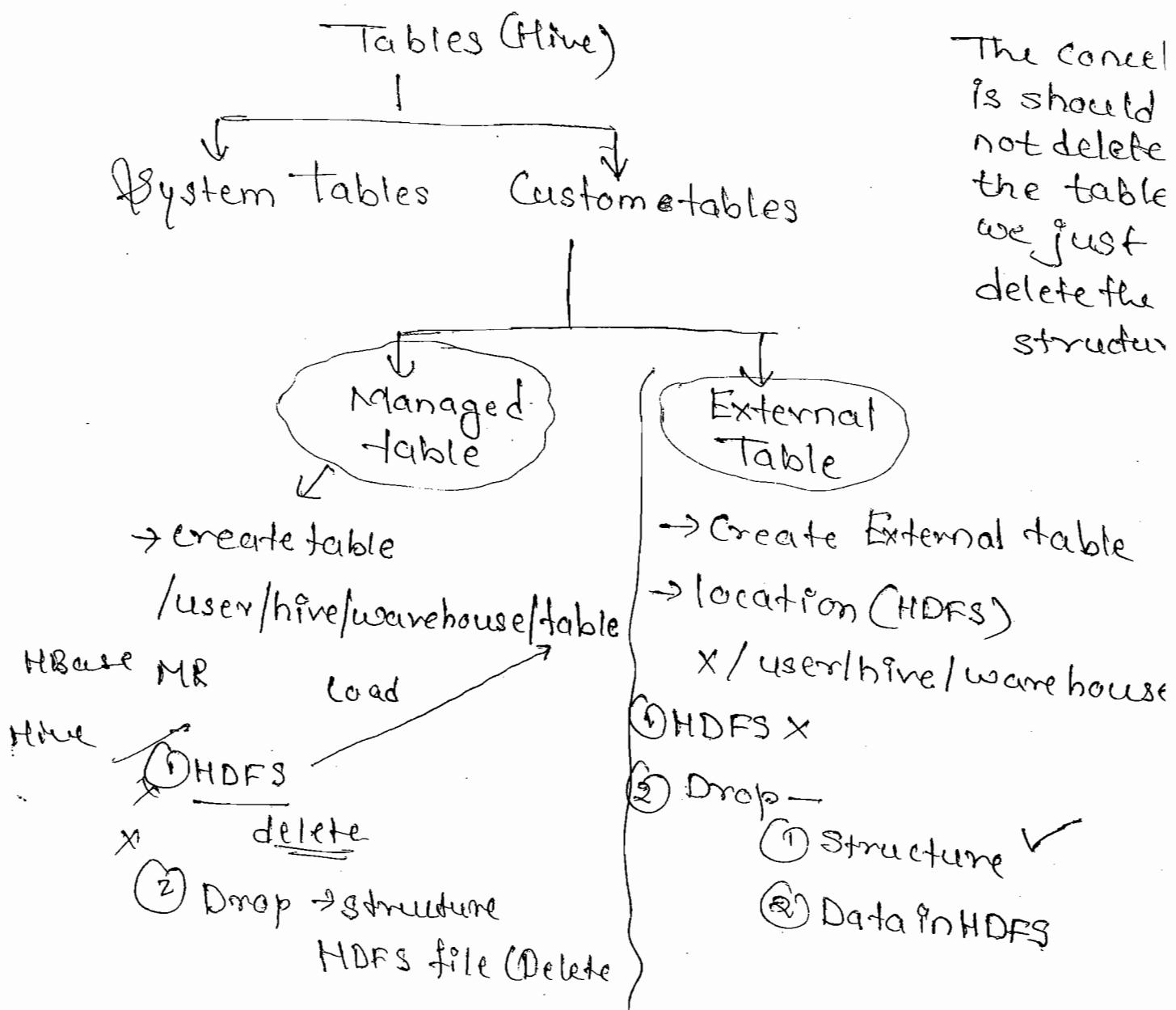
\$ cd \$HIVE_HOME

hive(default)> show tables;

hive(default)> select * from table records;

close Hive_lab2

In Hive, there are two types of tables



Goto hive – (when you come to Hive all are custom tables)

hive (default) > describe extended employees

→ look to the last row → tableType: MANAGED_TABLE

hive(default) > select * from employees;

Browser localhost:50075

↳ /user/hive/warehouse/employees/employee.txt

\$ mysql -u root -p

mysql > show databases

mysql > show tables

mysql > select * from TBLS;

hive(default) > drop table employees;

goto browser, there is no table name
employee.

mysql > select * from TBLS;

External tables → Hive_Lab2

Copy this file → and run it in hadoop.

\$ bin/hadoop fs -put /home/vm4learning/sam'

code/hadoop - - -

Goto Hive

Create External table records (year - - -

- - - 'ft' LOCATION '/user/vm4learning/
externaltables';

(hive(default))> describe extended records;

(hive(default))> LOAD DATA INPATH '/user/um4-
learning/sample.txt' INTO TABLE records;

Go to /user/um4learning/external-tables

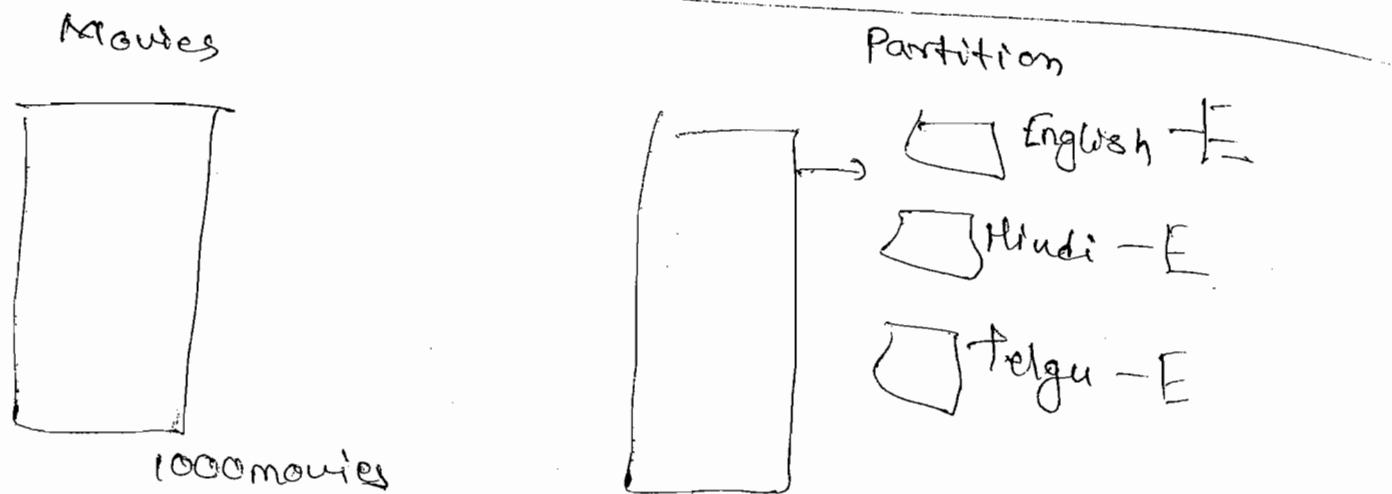
(hive(default))> select * from records;

(mysql)> select * from TBL;

(hive(default))> describe extended records;

← , → drop table records

Go to /user/um4Learning/external_table → HDFS



→ It is a partition therefore not is structure.

Data types } How
External tables } just copy paste

Home | Desktop | Finstone | inputs

Open → 2001-01-01-CR.txt

Goto partition in Hive-lab → Manage table

copy → Create Table logs(ts ^{BIGINT}, ^{SIMINT}, line string)

Hive (default) > paste it from partition 'create'

Hive (default) > describe logs;

Hive (default) > describe extended logs;

/user/hive/warehouse/logs

→ we don't have any data

Hive (default) > Select * from logs

Hive (default) > load data local inpath '/home'

vinilearnin | Desktop | Finstone | 2001-01-01-CR.txt

into table logs partition (dt='2001-01-01')

Hive (default) > select * from logs;

Browser go to /user/hive/warehouse/logs/ dt=2001-01-01

/country

hive(default)> select * from logs where dt='2001-

hive(default) _____ , - 01-01' }
and country='GB' and ts=123;

hive(default)> load data local inpath '/home/rmklearnig/
/desktop/firstone/inputs/2001-01-01-
India.txt' INTO TABLE logs Partition
(dt='2001-01-01', country='India');

/user/hive/warehouse/logs/dt=2001-01-01',
.

hive(default)> select * from logs ;

hive(default)> select * from logs where dt=

'2001-01-01' and country='India' ~~and ts=126~~

and ts=126 ;

~~9/18/15~~
~~Thursday~~

Hadoop Cluster → copy from Hive-LAB from external tables

bin/hadoop fs -put /home/vm4learning/Desktop/firstone/Inputs/hr-data.txt /user/vm4learning/hiveInput/hr-data.txt

Goto /user/vm4learning/hiveInput

Go to the Hive-Lab → from external tables

→ create

Goto Hive

paste it

hive default> create external table employees(

id int

default) select * from employees;

↳ describe employees;

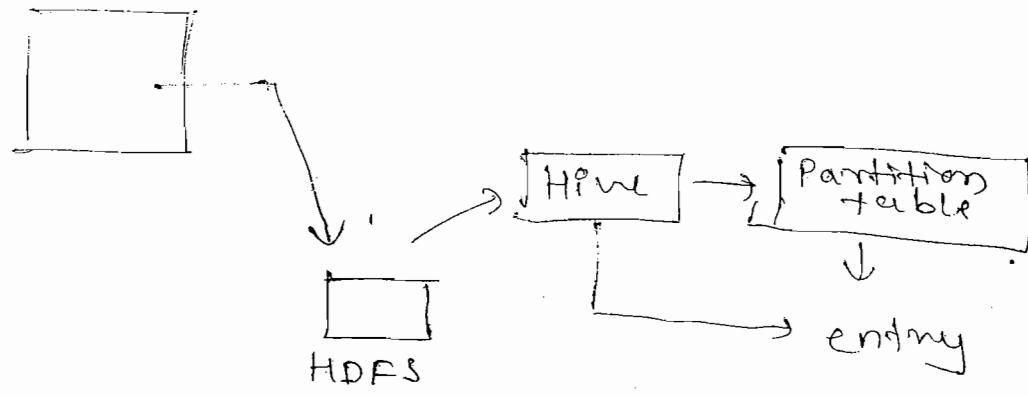
no-partition table

writing the O/P of one query to another

↳ table →

hive copy → select id, first_name, last_name, email,
(default) into
hive

hive (default) paste



How to load the table into another table
 or How to write the O/P of one to the other table.
 Now, we are going to make a new table
 called partition table

copy

↳ Create external table emphive (-----)
 hive(default) paste

Staging Table

hive(default) > Select * from emphive

Now go for next command from the same file

Insert overwriting table → copy

() > paste

> Select * from emphive where
 country = 'DE' and state = 'Bavaria'

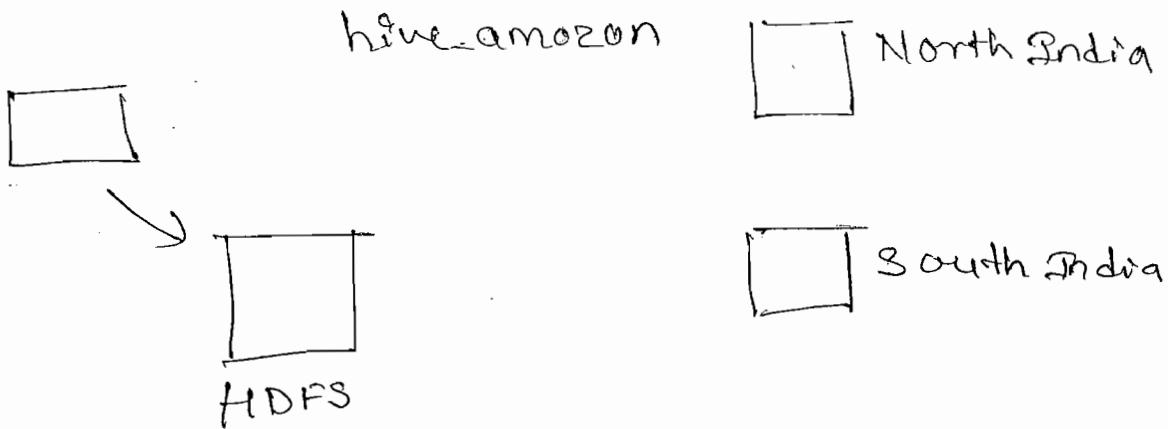
What are the types of table?

→ Managed

→ External

hive (default) > select * from empview

Hive is going to give you new concept
called Multiple Records ^{Inserts} _{...oo}
y these you do not need to take info single at a
MULTIPLE INSERT - time you can do multiple
 _{insertion}



We have 2 tables & we have to be insert multiple tables at a time.

~~Note~~
In a single scan we are going to insert data in two tables ...oo

MR E

Insert into NorthIndia
partition (city='India', st='1')
select * from Hive-amazon
where contr='India' & st,

Select
Insert we South put (contr=India
--)

Select * from Hive-amazon
where count & st

~~From Hive~~ External tables

Tables

~~Copy from IMP~~

Copy from Multiple Inserts -

we have Syntax -

from employee

insert into table -----

→ copy

hive(default) > select * from emphive;

hive(default) > show partitions emphive;

→ it will show all the partition value.

hive(default) > select * from emphive;

hive(default) > paste

> select * from emphive;

> show partitions emphive

> select * from emphive where
country = 'US' and state = 'Texas';

Disadvantage

avoid → (multiple Insertion is
Hard Coding)

If 1 country is having 3 elements

& 50 country — 150 elements

we have to write 150 elements for this
therefore No hardcoding to overcome this
we have —

hive (default) > drop table emphive;

hive (default) > create ← again create it
as previous.

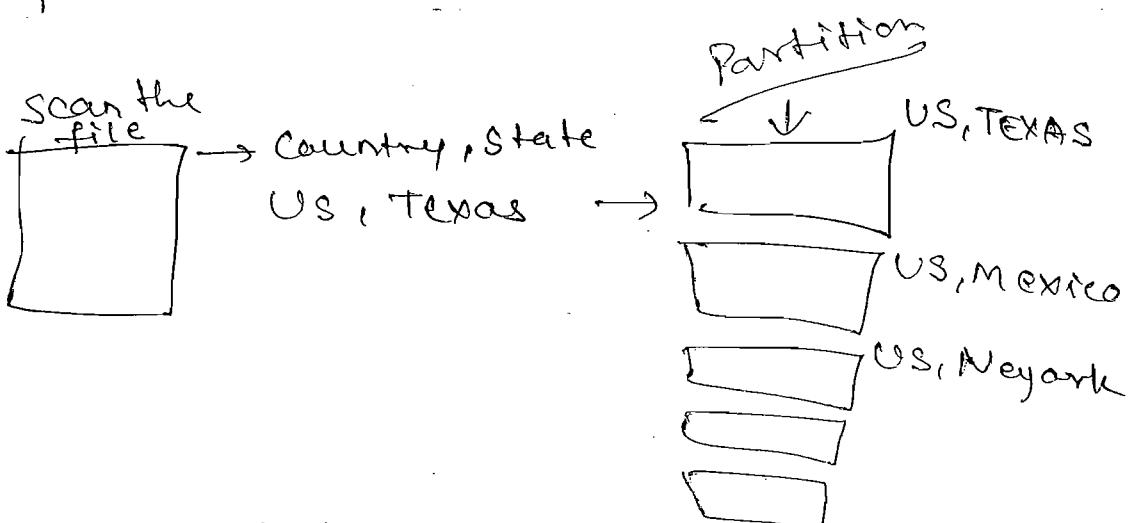
↳ show partitions emphive;

↙ (we do not have parti)

Now we do Dynamic Partition

Go to Dynamic Partition

copy all the things (5 lines) {set}
and paste it into the shell



paste it into Hive

hive (default) > insert ← next code

hive (default) > show partition emphive;

without dynamic partition, we want to do it

hive (default) > select * from emphive where country =
'UK' and state = '';

why
partition → for
Better
Performance.

BUCKETING

Partition → grouped column

→ Bucketing can happen on partition or unpartitioned table.

Sampling can not be done on partition for this you need new concept...

there are two types of Bucketing (use for sampling)

hive (default) > set hive.enforce.bucketing;

= false

hive (default) > set hive.enforce.bucketing = true;

> set hive.enforce.bucketing;

= true

copy → from BUCKETING

hive (default) > paste

> select * from user;

bin/hadoop fs -put ← copy

and paste into Hadoop(HDFS) ← enter

/user/vmlearning/hive-bucket

/user/hive/warehouse/user

Now

Load → copy and paste into hive

→ paste

→ select * from user

/user/hive/warehouse/user

→ describe extended user

there is no bucket because

there is no bucket as a table

Now we create

→ create table

→ select * from bucketed_users;

→ describe extended bucketed_users;

see → [bucketcols: [user_id]]

→ select * from bucketed_users;

→ select * from user;

→ INSERT OVERWRITE TABLE

→ copy & paste

→ select * from bucketed_users;

go to browse

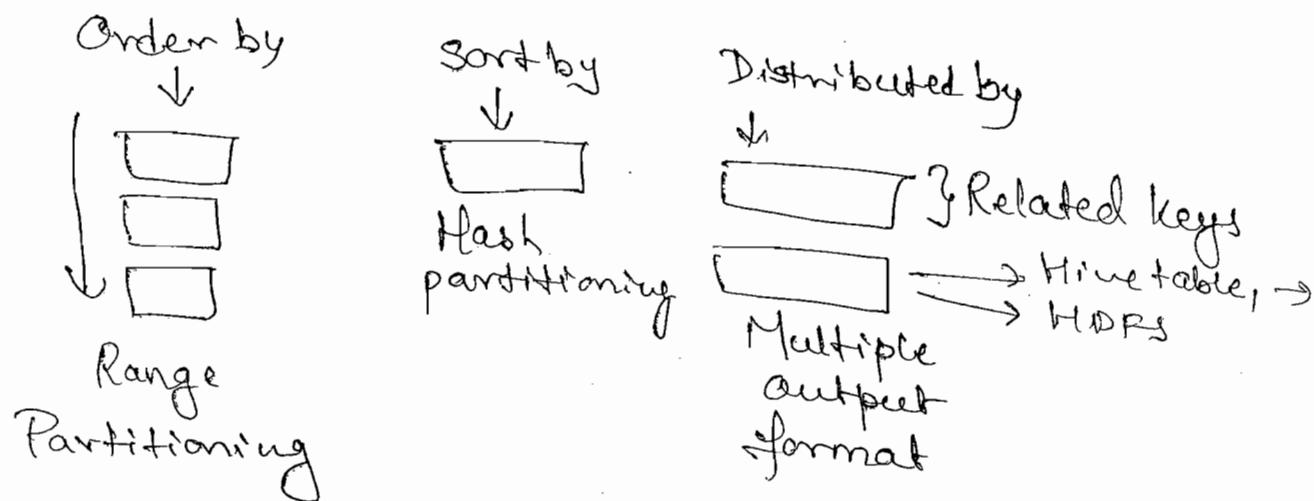
→ select * from bucketed_users

Tablesapple (.Bucketed¹ out of 2 ON user_id);
2 instead of 1

Hive - PPT

There are different "by's" in Hive →

- (1) Partitioned by
- (2) Clustered by
- (3) Order by → (In a sorted manner)
- (4) Sort by → (Sorting per reduce level)
- (5) Distributed by → (Sorting within reducer
& all related key comes together
& the O/P will be in sorted order)



All coding are present in Hive-LAB

Hive (default) > select * from employees;
 > copy & paste every line ↴

Browser → /user/vm/learning/distribute

50075

~~24/07/2018
FRIDAY~~

FUNCTIONS (Hive)

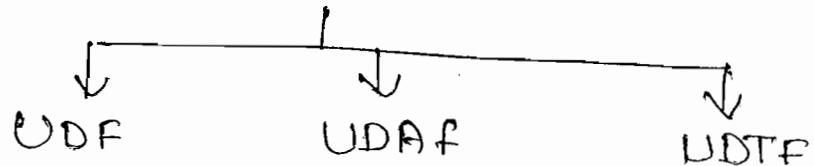
System Defined
functions

User Defined
functions

hive(default) > SHOW FUNCTIONS & Show system function

→ When the System function not having that we want , then only we goes to UDF

3 types of User Defined function



UDF
(User defined
function)

UDAF
(User defined
Aggregate
function)

UDTF
(User defined
Tabular
function)

- It always return single value
- It convert small letters into capital.
- Map only job and convert capital letters into small letters.

- It will return single value
- Map reduce job
- Taking Multiple column

- take single column & generate multiple column
- Map Reduce Job

Open Eclipse -

→ Open HiveUDF →

↳ src

↳ project called → Lower.java

↳ right click

export

↳ say jar file

↳ Desktop/HiveUDF.jar

↳ Done

Check it out, you got a folder called
HiveUDF.jar

Go to Hive-LAB

↳ User defined function

↳ Create Table statements → copy and paste
in in Hive

hive(default) > create table student

(default) > select * from table student

Load data local inpath → copy & paste in hive

hive (default) > LOAD Data Local Inpath --

hive (default) > select * from table student;

hive (default) >

Now I want to convert small letter name
into capital

copy the command → add jar → and paste
it into Hive

hive (default)> ADD jar /home/um4Learning/
Desktop/Hive UDF.jar ;

Now create a function

hive (default)> ~~SHOW~~ SHOW FUNCTION ;

hive (default)> Create temporary function Ragu
as 'com.example.Hive.udf.lower'

Now you can see Ragu as function

hive (default)> Select ragu(name) from students;

Goto Hive_lab

(\\$) bin/hadoop fs -put NOTICE.txt /user/um4learning/
output/NOTICE.txt

place this file in Hadoop

\$ paste the things

Goto
→ /user/um4learning/output → SOOTs
Browser

NOTICE.txt

Hive
 (default) external
 Create table if not exists }
 > Row format }
 > stored }
 > location }
 Copy &
 Paste
 from
 Tab2 Hive-Lab

Now copy the next ~~create table & an to~~
 and paste onto the Hive.

row-lines, word count

↙
 ↘
 & tables

newline
 latter
 creating
 ↓
 go for
 map-output

Run that part only

hive
 (default) > from row-lines

=
 =
 =
 -

hive (default) > select * from row-lines;

hive (default) > Run everything [from {

}] }
 the whole
 line now
 we have to
 paste &
 then say enter

→ you are only inserting
 the one table into the
 another table..

UDTF → User defined Tabular function

Go to Hive LAB

↳ UDTF

→ Distributed Cache

→ An approach used by
MR by the distributed
file across datanode

What are the files present in distributed
cache?

LIST CACHE

hive> LIST ~~CACHE~~ FILES; [or LIST JAR;]

hive(default)> LIST FILES; ← enter

Explode() → convert complex datatypes into
primitive datatype for that
we have a tabular function
called Explode();

Goal

→ Hive LAB

→ UDTF

→ copy "create table employees -- "

> drop the table first called employee

} and then paste it into the Hive
create line

Hive(default)> select * from employees;

(default)> Load data local inpath

(default)> select explode(subordinates) as names
from employee;

Disadvantage - of (explode())

→ we cannot select another column; (explode())
→ UDTF are not support outside the Select
clause, nor nested

→ takes Array as input

→ cannot do nested

→ No other expressions are allowed.

Hive Select explode

(default)> select name.salary -- As Names; → copy

Hive(default)> paste the line ↑

- Now again copy the next code

Hive
(default)> select name, salary

Lateral View

← enter it

view the capital letter in
small

RCFILE DEMO: → Hive-LAB

How

→ All the codes are already present, we have to only copy paste and run it.

Goto → /user/hive/warehouse/user.txt → Browser
→ 50075

→ To see ← 50030

MapReduce
job

→ Instead of RCFile we can stored as

a sequential file too;

NOTE →

→ select * from RCFILE
^{line}
(default) describe extended USER_RL;
^{line}
(default) select * from user-text;

Goto RC file whenever we
want to do aggregation
we go for RC file.
→ It will do the process
faster.

earlier it is
in row fashion
now we get
the data in
column fashion.
You can see the
change only in
browser not
on hive
→ if you write
select * from
USER_RL

~~MAP SIDE JOIN~~ → HDFS → Hive LAB

Home Work → do all operation at Home

How to Run Hive script without login

Jump to the Shell &

↳ Installation

↳ hive-0.11.0-bin

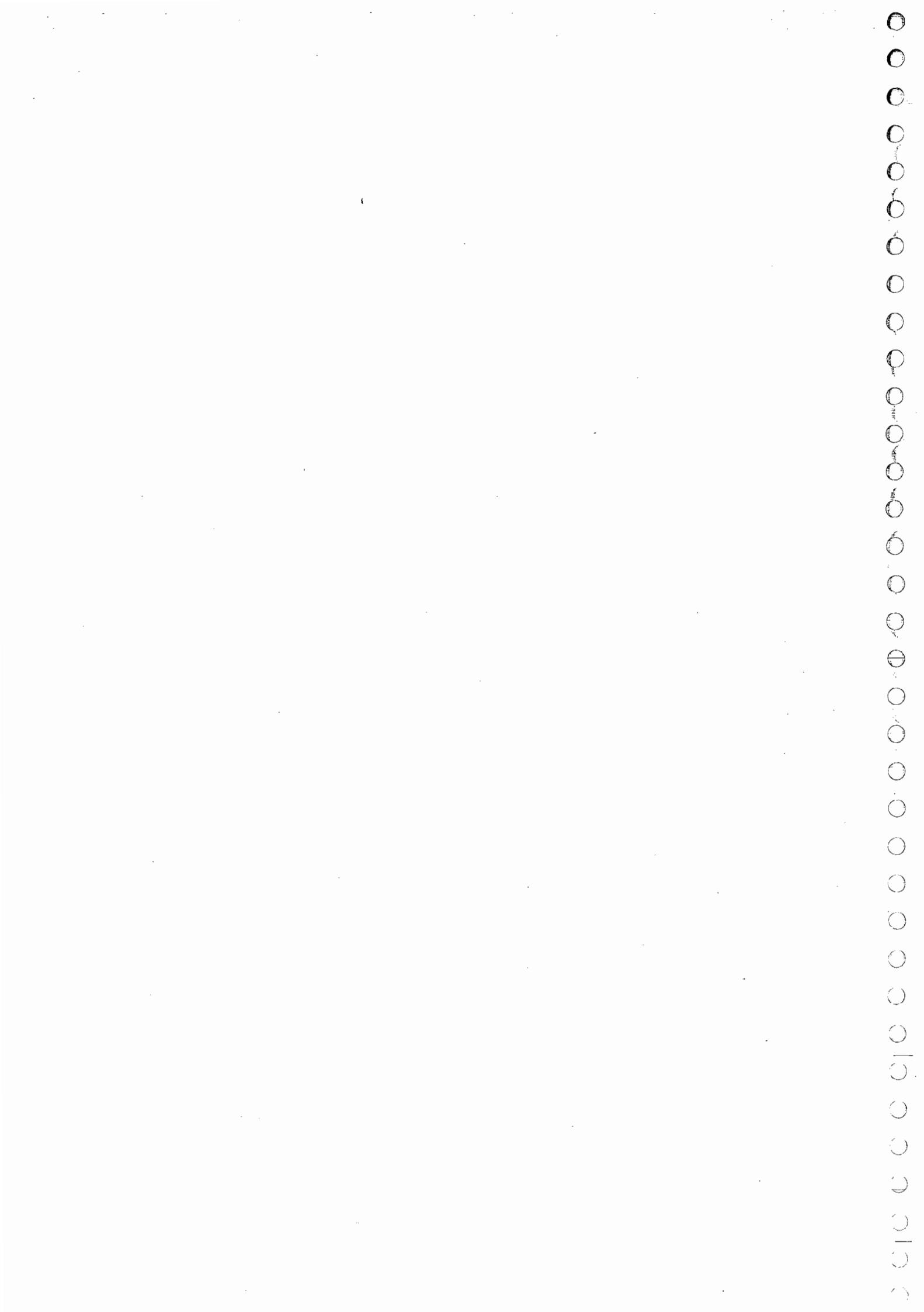
↳ scripts → create OurScript

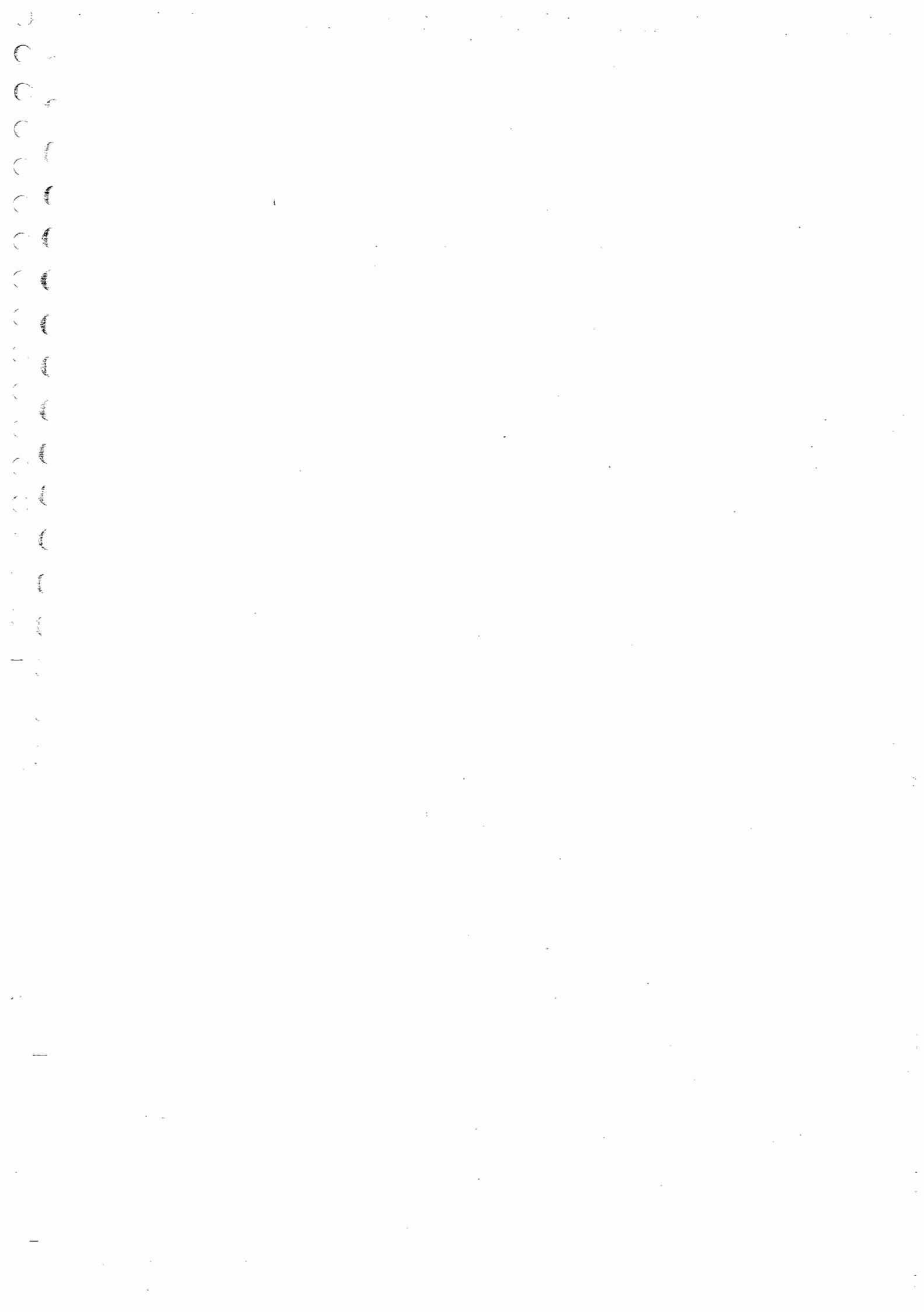
↳ OrderBy-~~text~~.txt

Pseudo - Create a VM

- ① Install Java into the mission. t:6 (Java 1.6)
- ② Create User → localhost
- ③ key & SSH
- ④ Download Hadoop
google it → hdfs site.xml

PUTTY 8



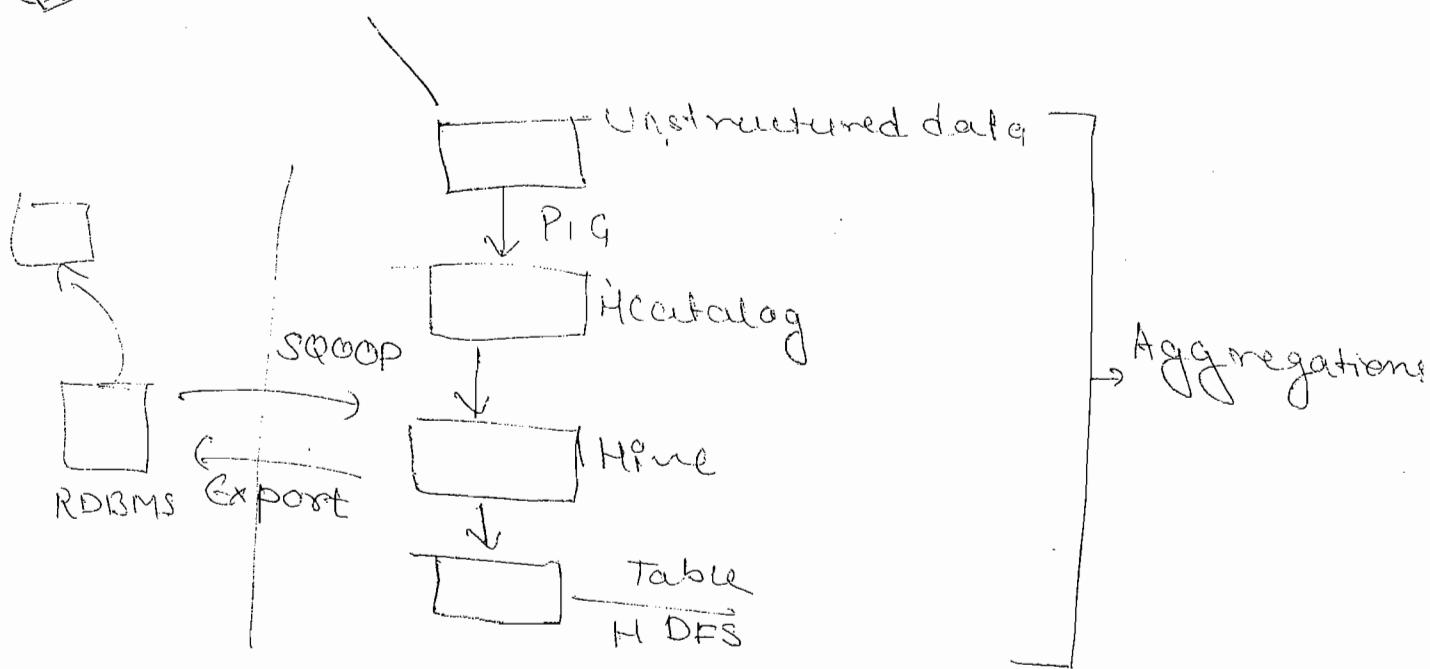


3Q00P \Rightarrow

26/07/2015
SUNDAY

→ This is totally Bash

SQOOP

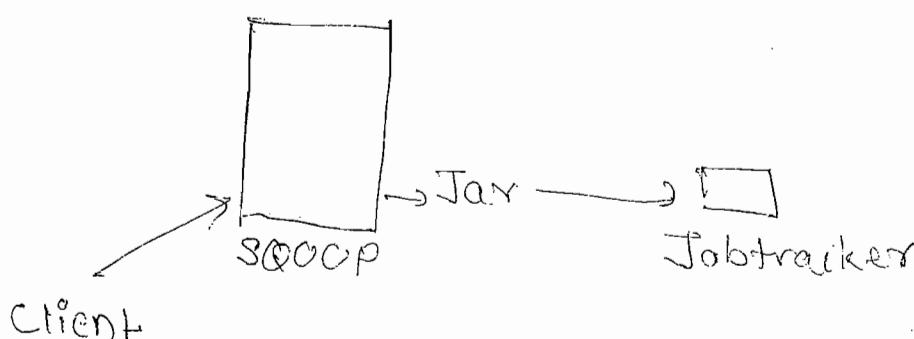


⇒ We are processing the data not predicting the data

- RDBMS takes Hadoop data
- For this we need structured data tool (SQOOP) for the purpose of transferring the data.

SQL to hadoop → SQOOP

→ You can inject the data into HBASE by RDBMS.

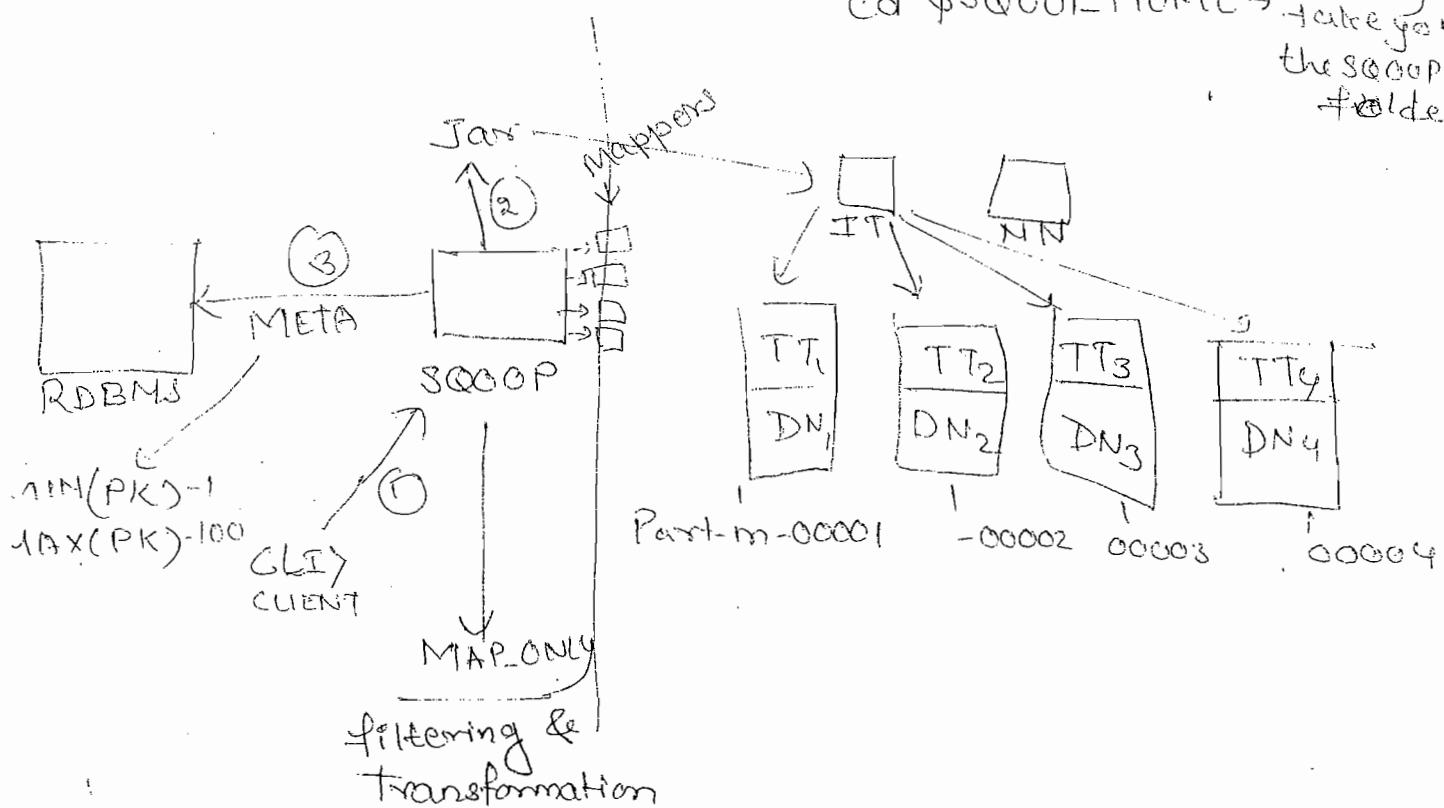


TBLS

↳ Primary-key (PK)

bin/start-all.sh

cd \$SQOOP_HOME → directly
take you
the Sqoop
folder



What are the things require?

- (1) Connection Required Stream
- (2) HDFS folder

→ Sqoop will generate corresponding jar file & this jar file submitted to the Jobtracker.

→ Sqoop will launch defaultly 4 Mappers

Ex → If I have 100 records & Sqoop is having 4 Mappers then $\frac{100}{4} = 25$ for 100 records it will take 30 min.

I know it can be done in 7.5 mins.

\$ bin/start-all.sh
\$ cd \$ SQOOP_HOME
\$ bin/sqoop help
\$ mysql -u root ->

mysql> show databases;

mysql> select * from tbs

mysql>

In Sqoop home →

bin/sqoop import --connect jdbc:mysql://
localhost/hcatalog --table TBS -m 1

Goto mysql:-

mysql> grant all privileges on hcatalog.* to '@'localhost';
mysql> grant all privileges on hcatalog.* to '%' '@'localhost';

/user/vm4learning/TBS

HDFS Browser

We can see New TBS will be shown in browser

→ We are giving some privileges to our home.

Hadoop-

\$ bin/hadoop fs -rmr /user/vm4learning/TBS*

Deleting RDMs

Tab → | HADOOP | SQOOP | MySQL

SQOOP →

\$ bin/sqoop import --connect jdbc:mysql://localhost/hcatalog --table TBLS -m 1
/user/vm4learning/TBLS

HOME | FILESYSTEM | Temp | SqoopVm4learning | compile
TBLS.java

Again delete the TBLS folder

Come to Sqoop

→ Now again connect by the code

Now we are going to start sqoop

Sqoop is a client tool.

Hive ——————

Pig ——————

} No need to
start cluster.

sqoop

\$ bin/sqoop help

→ Just to know the

mysql> show database

connection is working
or not fine or not

List - databases → :

sqoop \$ bin /sqoop list-databases --connect jdbc:mysql://localhost/hcatalog

SQOP \$ -- list-table --

SQOP \$ show-table

Remove the folder TBLS from HDFS again

mysql> select * from TBLS

sqoop \$ bin /sqoop import --connect jdbc:mysql://localhost/hcatalog --table TBLS --where "SD_ID = 63" --target-dir /user/vm4learning/TBLS

SQOP \$ bin --

-- /hcatalog --table TBLs --target-dir

/user/vm4learning/Namesh

* user/vm4learning/Namesh

bin | SQOOP import

- connect
- table
- m
- options
- where
- target-dir
- split-by
- Query

mysql> describe TBIS;

SQOOP>target -b, split -b

(① & nd Query code)

\$ bin / sqoop import -- -- -- -- --

--fetch size -5000 --split by SD.ID ← Run it in sqoop

remove this line

& run

27/07/2015
MONDAY

\$ bin/hive --service metastore &
hive 0.11.0

→ This 2 lines will be there
in folder

↳ Ourscript

↳ Orderby-Test-Sql

\$ bin/hive (shell)

(hive) select * from employees;

(default)

INSERT OVERWRITE DIRECTORY

'user/vm4learning/Input1'

select id, first_name, country,

state from employees order by

country asc, state asc;

hive (default)>

External

Without login to the shell

~~Shell~~

How we can see what is there
in table?

\$ bin/hive -e 'select * from employees';

→ Without login to the shell, we can run query..

Orderby-Test-esq;

↳ right click

Properties → copy the location

Table → Table
 Directory

/Inst/hive-0.11.0-bin \$ bin/hive -f /home/

vm4learning/inst/hive-0.11.0-bin/ourscripts/

orderby-Test.sql

If you do not have employee table

then go to Hive-Lab

Go to External table

↳ Create external table → copy the last created of the notes
and paste it into the Hadoop cluster

Hadoop
\$ paste

Go to

/user/vm4learning/InputL

Compress on HiveTable → Hive-LAB

\$ -mrv /user/vm4learning/InputL

Go to /user/vm4learning/InputL

Hive ~~conflict~~.conflict is the thing which should pass parameter to shell.

How you pass \$ dynamically to the location?

Goto - hive-lab

Dynamic Substitute Values in hive

hadoop bin /hive → 1st line paste and run it here...

Goto →

/user/vm4learning/dynamic1

-bin > orderby.sh

Running Shell Scripts

\$./orderby.sh

Goto /user/vm4learning/Input → we get our data back

chmod +x *.sh → all sh got execute permission

our scripts > \$./Dynamic.sh '/user/vm4learning/Dynamic-output/' ↪

→ Your code should be dynamic not hardcoded

Goto → /user/vm4learning/ =

Q We can delete a single record of the table or not?

→ Yes we can delete by the use of overwrite.

→ Insert overwrite into table students
select * from students where age < 5

Raghav	5	1000
Ravi	25	3000
Raju	30	5000

After query
we have
only two
record...

functions →

① Row_number →

② Rank

③ DenseRank

Q How to get the second highest salary?

→ Row_number() over (order by sales_report)

Q How to get the 1st highest salary?

→ order by → [select top 1 sal from emp
order by salary desc]

Q - Get me the 3rd lowest no. ?

⇒ DENSE-RANK(sales_amount) over (order by sales_amount)

			Rank	
A	X	100	1	1
B	X	200	2	2
C	X	200	3	2
D	X	500	5	2
E	X	400	8	5
F	Y	500	6	8
G	Y	300	7	5
H	Y	300	9	5
I	Y	500	10	10
J	Z	200	4	9
K	Z	700	11	7

Goto → HIVE2.adox → Imp

We have web console for Hive too

⇒ HIVE Web Interface

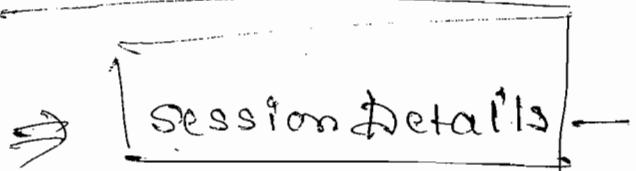
(HWI)

\$ cd \$HIVE_HOME

\$ bin/hive --service hwi

localhost:9999/hwi ⇒ Browser

Hive Web Interface



Result file → /user/tum4learning/output_hwi

Error file → _____ /output_error

Query →

Start Mode → NO

Start Query → NO

Submit

Session Name → First Session

"Running"

It will show the details
of your query
working

Database

Default table lists

Bucketed user

→ Goto this

You can directly write your queries
no need to go into shell...

~~28/07/2015~~
~~TUESDAY~~

BOOKS WHICH ARE NEEDED -

For HADOOP →

- ① Hadoop The definitive guide → 4th or 3rd ..
- * ② Hadoop in practice ..
- ③ Hadoop in Action ..

For HBASE -

- ① HBASE THE DEFINITIVE GUIDE
- ② HBASE IN ACTION

For HIVE -

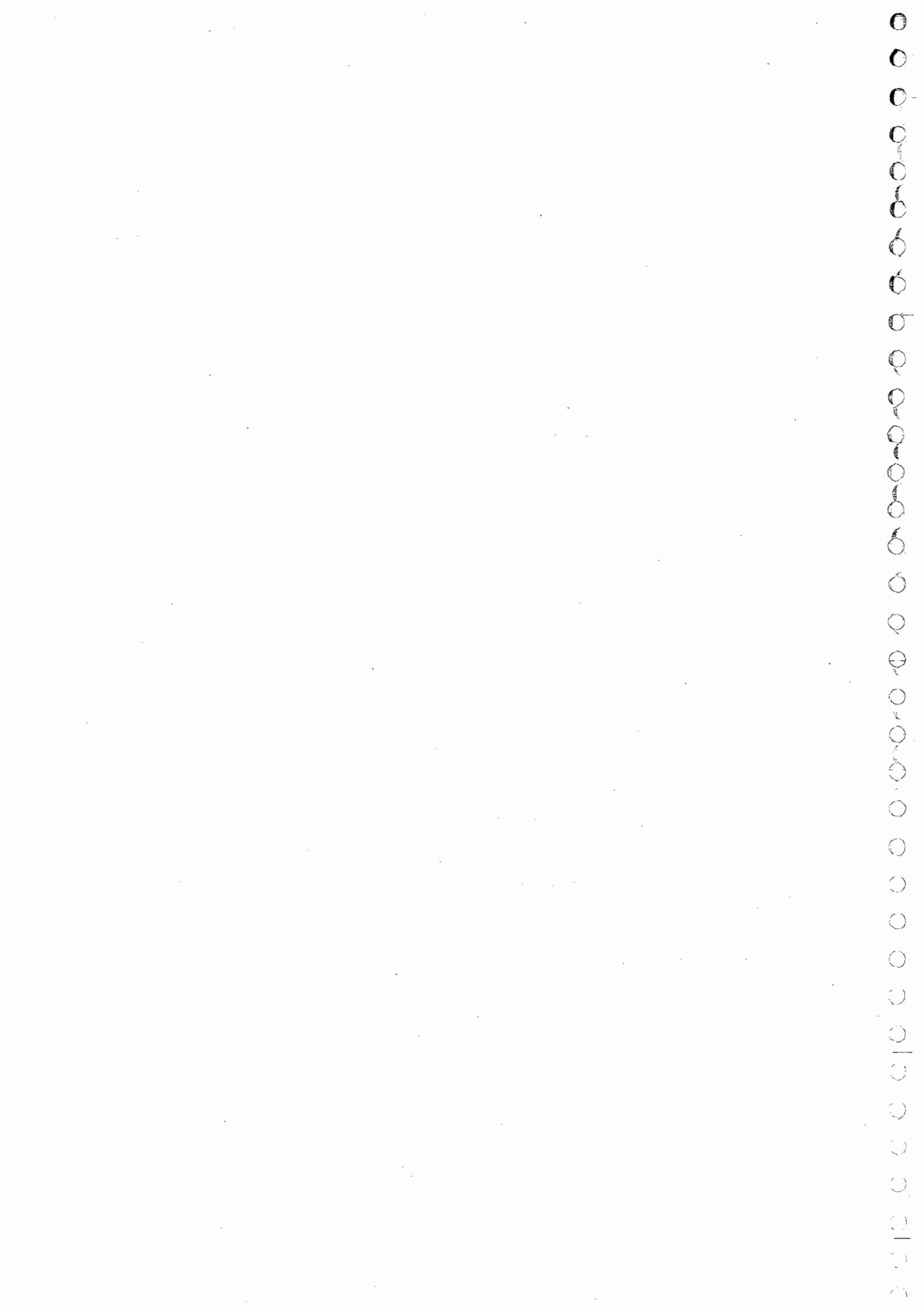
- ① Programming Hive

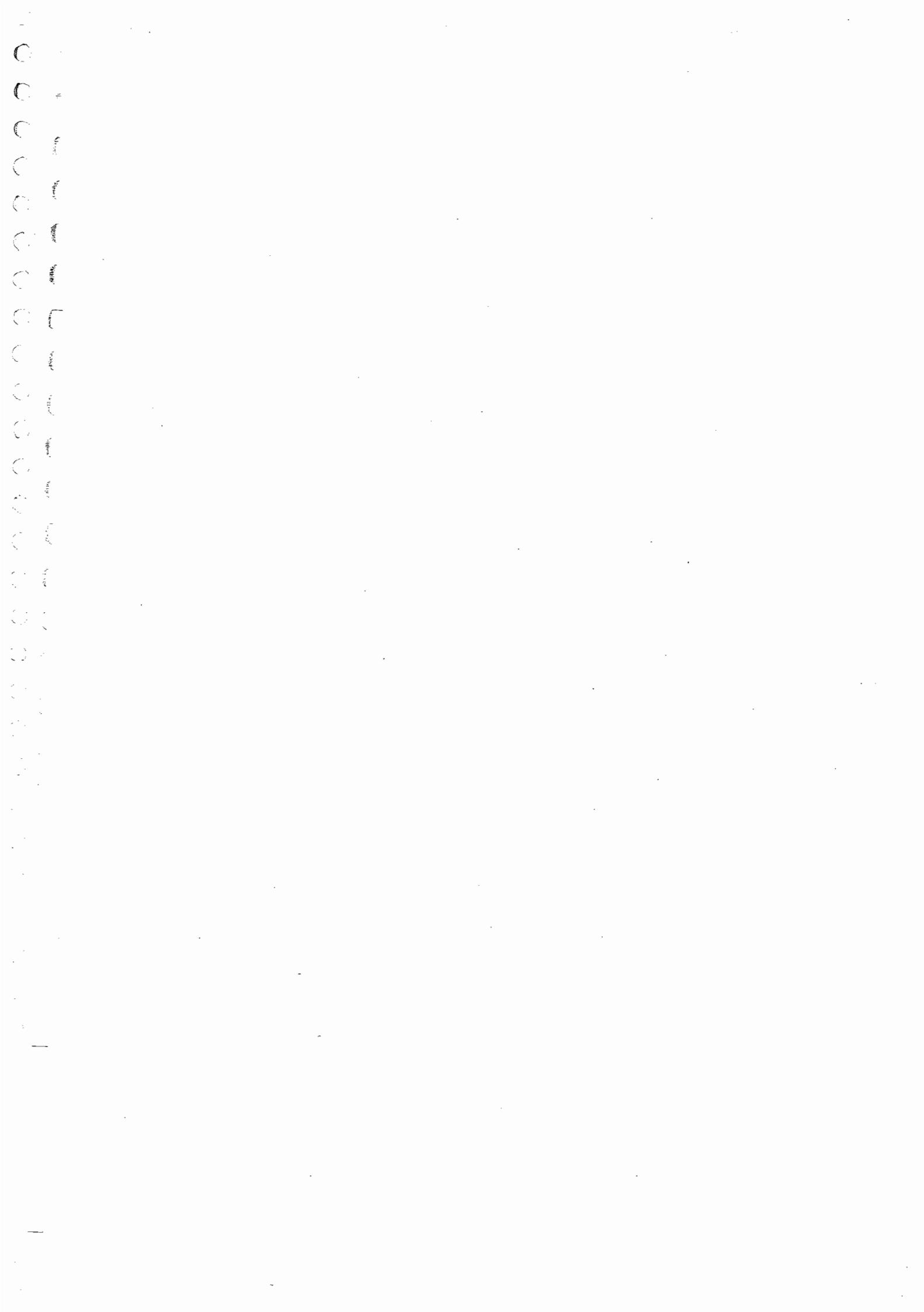
For PIG -

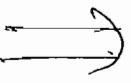
- ① Programming PIG

For Kafka

{ For BigData
① I ❤ Logs





PIG 

~~28/07/20
Tuesday~~

PIG

→ PIG was given by YAHOO

→ PIG also resembles SQL

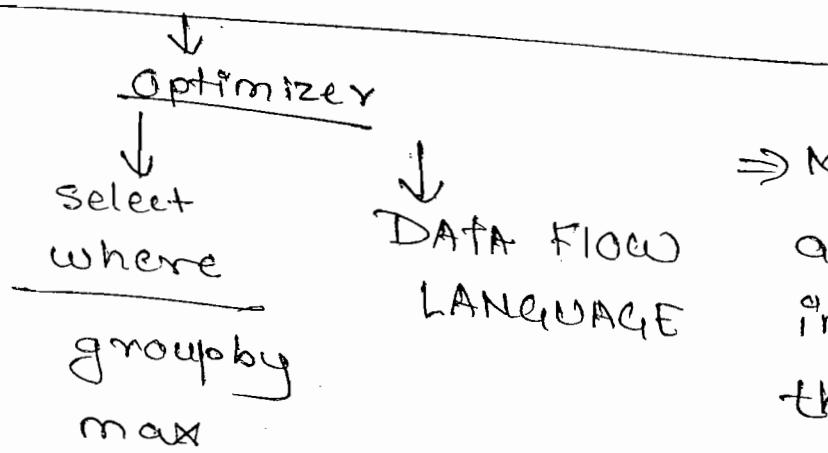
Now, we see that Why we go for PIG?

What we find difficulties in HIVE ?

Select ① max(temp), Year from record

where ② temp < >

groupby year;



⇒ Multiple joins
are hard to
implement in
this

→ And for this we need a language or
frame work called PIG... .

- PIG works for unstructured data too.
- PIG is act^{as} a preprocessing tool or you can use processing tool.
- PIG won't have data structure.

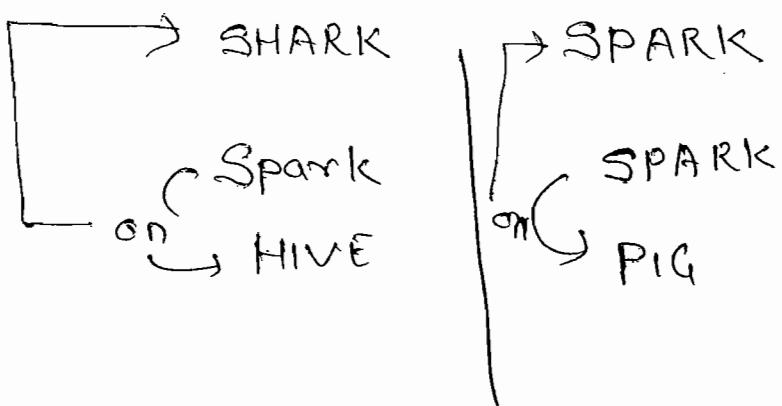
- PIG uses a language called Piglatin
- PIG has a shell called Gruntshell
- Hive & Pig are interactive shell
- PIG is also client tool
- Client tool → MR plan → Jobtracker

What happens in Update → Select requirement
Delete previous
Insert new record.

Complexity :-

It lies between Hive and MapReduce

Hive → PIG → MR



Goto Hadoop-HOME

\$ cd \$PIG-HOME

\$ pwd

\$ ls

Goto Installation

↳ PIG-0.12.0

↳ bin

\$bin/pig → This will log you into the
Grunt shell

grunt>

Why pig is going to run the task fast?

⇒ Because Pig is lazy...

Let we place a file in HDFS

Open → PIG-LAB

Executing the pig script to find the max
temp for a particular year →

\$ bin/hadoop - - - - - sample.txt

copy and paste in Hadoop cluster
/user/vm4learning/pig-input1

Here Relation is nothing but a table in Hive.

PIG-LAB -

Execute the below commands →

records = load 'pig-input/simple.txt' AS
(year:chararray,temperature:int,quality:int);

→ Goto pig shell and paste it...

Now Do you think file got loaded...

No it is not loaded...

Because PIG is lazy...

→ To load the record; we need to use

do dump records...

grunt> describe records;

grunt> dump records; ← it shows the
actual record...

⇒ It will perform action or Read not
at the time of write because of lazy
evaluation...

→ If you want to communicate use
simple english

Goto \rightarrow PIG-LAB

Execute the below commands -

filtered records = FILTER records BY $\circ\circ\circ$ copy

localhost = 50030

grunt > paste here

grunt > dump records;

Now copy 3rd line

grunt > grouped_records = GROUP filtered_records by
year;

grunt > dump grouped_records;

grunt > max_temp = FOREACH grouped_records GENERATE
group, MAX(filtered_records, temperature);

grunt > dump max_temp;

grunt > explain

Come out of the shell & say again

\$bin/pig

grunt > dump max_temp

copy all the 4 retⁿ at a time & paste it into
the shell $\circ\circ\circ$

grunt>dump max-temp;

grunt>explain max-temp; → it will show
you records...

grunt>

A = LOAD

B = FILTER A

C = LOAD

D = ALTER C

E = group B

F = max E

G = group by D.

dump F;

Goto the PIG-LAB

You can Run Pig w/o Hadoop

bin/pig → MR

bin/pig -x local ⇒ w/o Hadoop you can login

Netflix Lipstick →

Putting ~~the~~ lipstick on Apache Pig

\$bin/pig -help → if you want to know something...

\$bin/pig

look the timestamp & INFO

\$bin/pig -help

\$bin/pig -b /brief login no timestamp

\$bin/pig -help

\$bin/pig -d (debug → how the flow is going on)

→ we can see by -d

\$bin/pig

grunt>set debug on;

grunt>set debug off;

grunt>records = load 'pig-input' ---

grunt>set debug on;

Come out of the shell

\$ cp log4j.properties.template → Not working
leave that file as

```
$ cd conf  
$ cd $PIG_HOME  
$ bin/pig
```

grunt > LO \Rightarrow LOAD LONG

grunt > FILTER

Come out of shell and say ls

```
$ ls  
$ bin/pig }  
grunt > register ...  
grunt > register ...
```

Now you can see direct result

inst/pig 0.12.0 | conf | autocomplete.txt \downarrow create file

'pig-input1/simple.txt'

grunt > records = LO

grunt > records = LOAD 'pig-input1/simple.txt'

grunt > records = LOAD 'pig-input1/S' $\downarrow \rightarrow$ shorthand

grunt > dump records;

grunt > describe records;

grunt > records = LOAD 'pig-input1/simple.txt'

AS (year:chararray, temperature:
quality:int);

grunt> describe records;

Default datatype of Pig is → bytearray

grunt> records = load 'pig-input1/sample.txt'

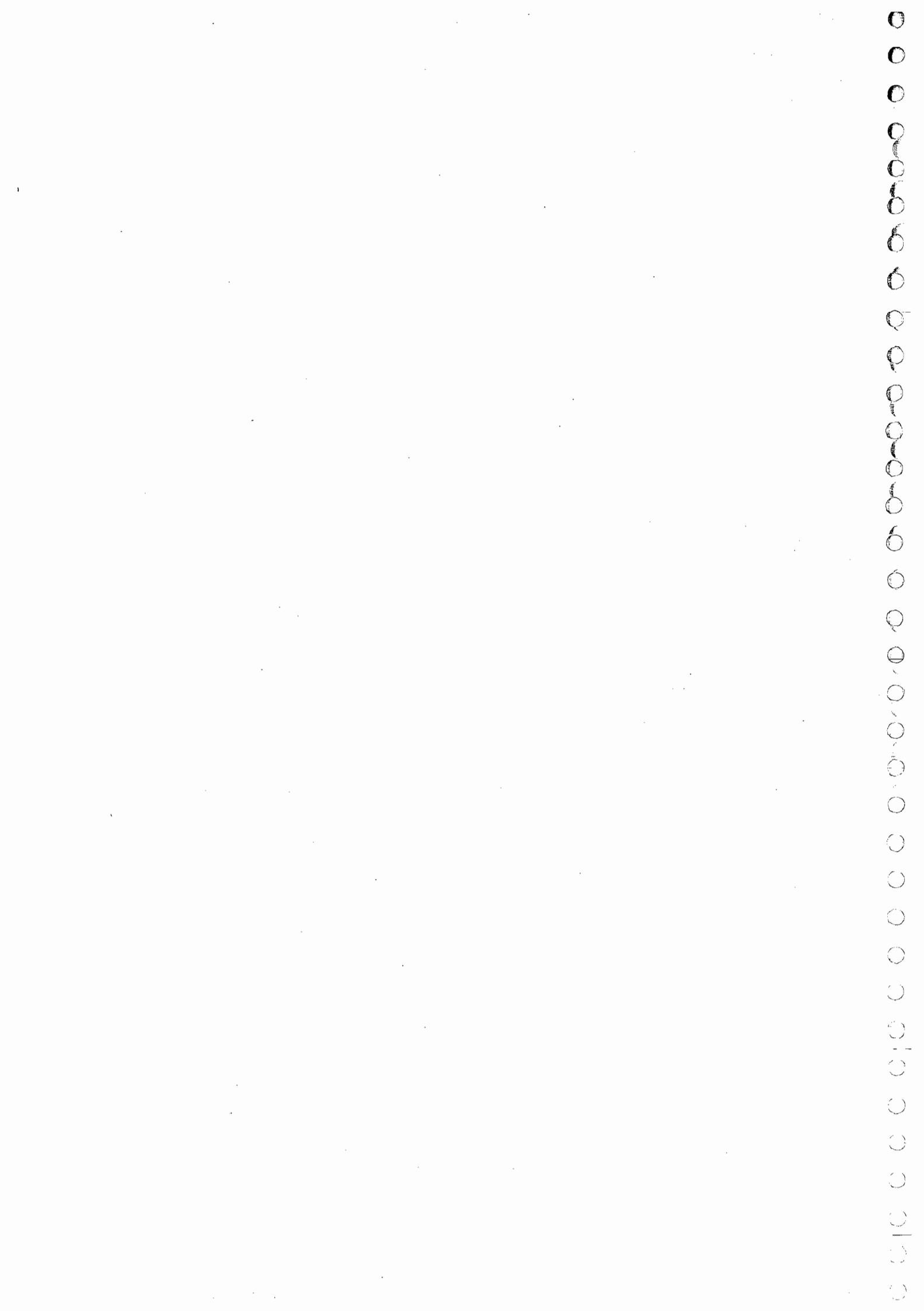
AS (year, temperature, quality,

grunt> describe records

grunt> records = load 'pig-input1/sample.txt',

grunt> describe records;

schema for unknown records..

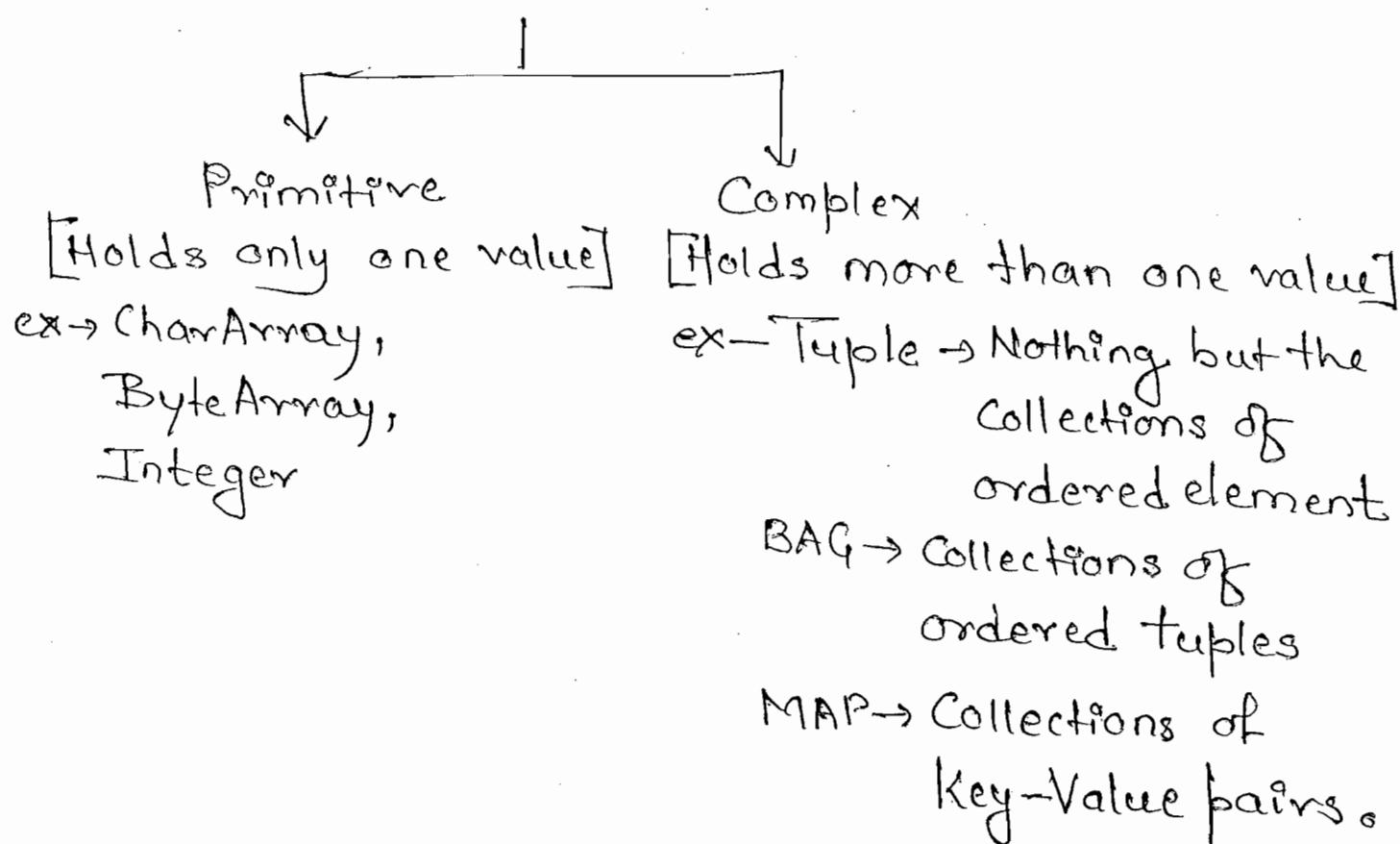


~~30/07/2015~~
~~Thursday~~

→ PIG also have complex datatype.

→ We have language called PIGLATIN.
very much similar to SQL.

PIG LATIN



```
$ cd $PIG_HOME
```

```
$ bin/pig
```

All Debugging Comments
grunt> Explain max-temp;

grunt> Illustrate max-temp; it will going
to do on the
subset of data

DUMP → will process the data → display on
Screen

DUMP → process - display
↓
used for testing purpose

grunt> dump max-temp;

STORE → Process → FILE(HDFS)

Store is process and stored into the
file called HDFS.

grunt> STORE max-temp into '/user/fm4learning/
max-tempfileop.txt';

grunt> STORE max-temp into '/max-tempfile
op.txt';

→ You can give a shortest path

Explain
Illustrate
Dump
Store

Very Important comment

grunt> fs

- It will directly give all the
- fs command will work in grant shell to

hadoop bin/hadoop fs -put /home/vm4learning/
PublicDatasets

grunt> fs - put /home/vm4learning/Desktop/
Firststone [Inputs/load-data.txt/user]
vm4learning / BigInput-all / load-~~txt~~ data.txt

Complex data type -

D = load \Rightarrow (copy this line)

grunt> D = load \Rightarrow paste the whole line here

grunt> describe D;

grunt> x = foreach D generate m#`city` as
cityName: character

grunt> describe X;

grunt> dump X;

Here we did convert the complex datatype into simple datatype...

TUPLE DATATYPE -

Complex datatype - (from PIG-LAB)

Copy 1st line to hadoop & pig...

Goto → user/vm4learning/piginput_all/
complex-data.txt

A = LOAD '/user/vm4learning/piginput_all/complex-data.txt'

t₁: tuple (t₁a:int, t₂a:int ; t₃a:int)

t₂: tuple (t₁a:int, t₂a:int ; t₃a:int)

t ₁	t ₂
3 8 9	3 8 9
1 4 7	1 4 7
2 5 8	2 5 8

You can access by
projection

grunt > A = `{ paste here }`

grunt > describe A;

grunt > X = FOREACH A GENERATE b1.t1a,
T2.\$0;

grunt > describe X;

X

BAG :-

BAG is Nothing but collection of
Tuples.

Goto

BAG Schema :-

bin/hadoop fs -put ----- copy and
----- paste into
----- hadoop

Goto: /user/vm4learning/piginput/all/
complex-data3.txt

{()}

PIG-LAB -

Goto → Validation in PIG

paste the file directly in grunt too

grunt > fs -put / - - - paste here - - -/

Goto -

/user/vm4learning/pig-input/sample-corrupt.txt

```
grunt>records = LOAD '/user/vm4learning/  
pig-input/sample-corrupt.txt' AS  
(year: chararray, temperature: int,  
quantity: int);
```

```
grunt>dump records;
```

→ If there is miss-match happens,
it will not show that value...

Q → How will we separate bad or
good records?

which is bad record?

⇒ which is having NULL value...

records → BAD - filter records by temperature
= NULL
→ GOOD - filter records by temperat-
ure < > NULL

Dump good records; } this is disadvantage

Dump bad records; } it scans the record
two times

Multi Query Execution; → Pig will do only
one time execution

{
 ↳ Query Execution } SPLIT records INTO GOOD-records If
 temperature > NULL, BAD-records If
 temperature = NULL

↳ this concept is same as multiple Insert... .

↳ By default pig is having multi Query Execution . . .

{
 ↳ grunt } SPLIT records INTO good-records If --
 -- -- --

↳ grunt } DUMP good-records ;

↳ grunt } dump bad-records ;

TYPE CASTING →

→ Converting one datatype into another
is called type casting..

grunt> A = LOAD '/user/vm4learning/piginput-all/missing-fields.txt'

Invented
↓
Yahoo → PIG

Google - HIVE

Goto → /user/vm4learning/piginputall/missing-fields.txt

⇒ The data consist is of type casting..

→ TOBAG

→ TOKENIZE

grunt> describe A

if Now, we are going to talk about
joins..

→ Pig is also have multiple joins.

Copy the first two lines of JOINs IN PIG
and paste it into the mission

Copy A = -- }
B = -- } and paste in Pig (grunt)

Q. What is meant by joins?

→ If we want to join two files

⇒ Both the file should consist same
columns

→ Select * from A JOIN B ON A.\$0 = B.\$1
grunt> C = JOIN A BY \$0, B by \$1;

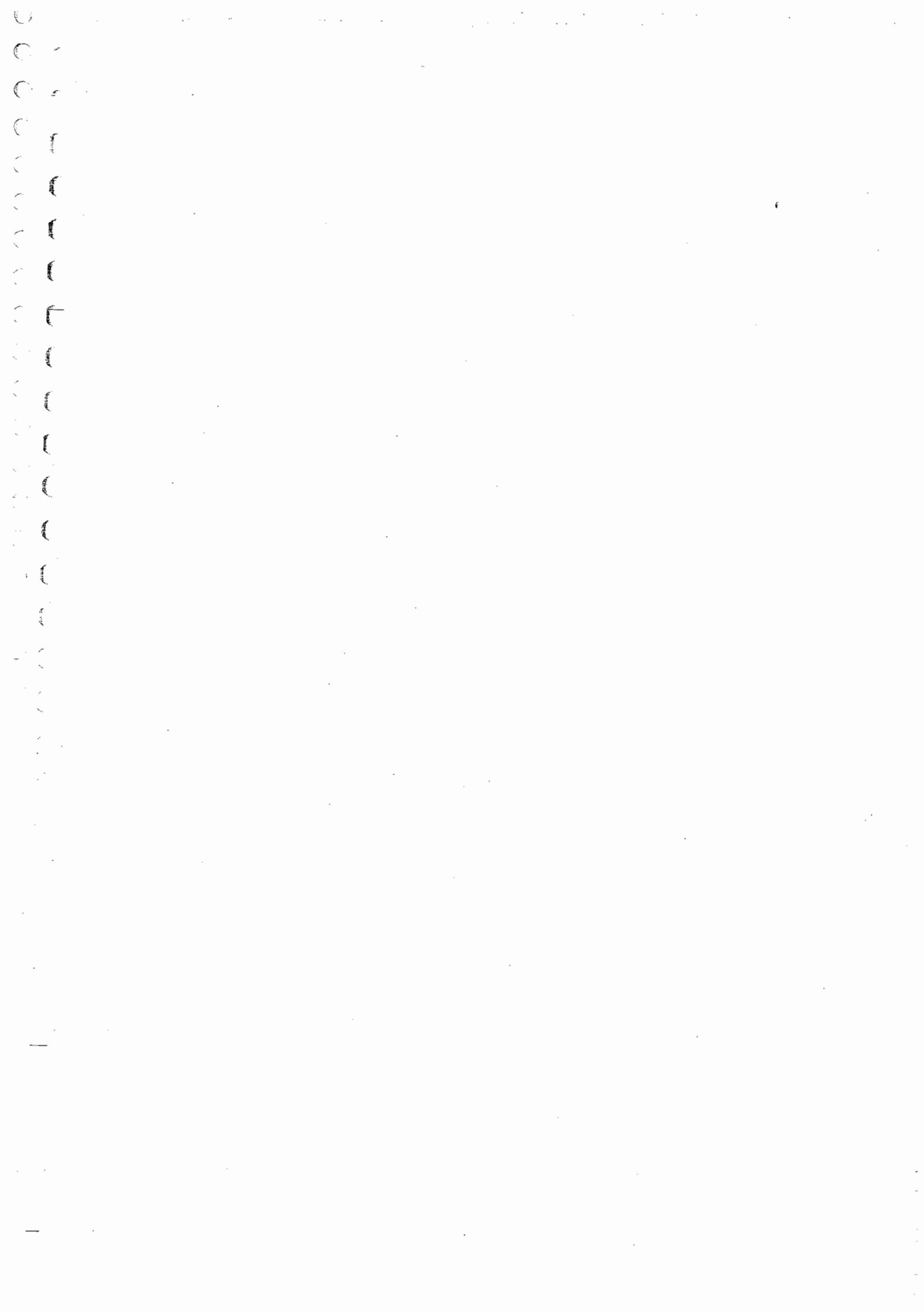
grunt> dump C;

grunt> D = FOREACH Generate C.\$0, C.\$1;

grunt> dump D; Important Functions

LOAD	SPLIT INTO
FILTER BY	JOIN BY
GROUP BY	TOTUPLE
FOREACH	EXPLAIN
GENERATE	ILLUSTRATE
	PUMP
	STORE

What is COGROUP ?



HBASE \Rightarrow

\$bin/hadoop fsck -move {destination}

→ It moves all the corrupted data to the destination

27/06/2015
Saturday

HBASE

— X —

Database

↓
SQL

↓
NOSQL

Schema on
write

(is a disadvantage) $\xrightarrow{\text{added}} \text{features}$

low latency

SQL follows ACID properties

ACID

→ Atomicity

→ Consistency

→ Isolation → Deadlock situation

→ Durability

NOSQL follows BASE property

→ BASIC

→ Availability

→ Softness

→ Eventual Consistency

Brewer's CAP theorem:-

→ C - Consistency

→ A - Availability

→ P - partitioning (distributed in nature)

In this world, there is no database which having all the three properties -

SQL \rightarrow Sequential Query Language

So, we can use in combinations -

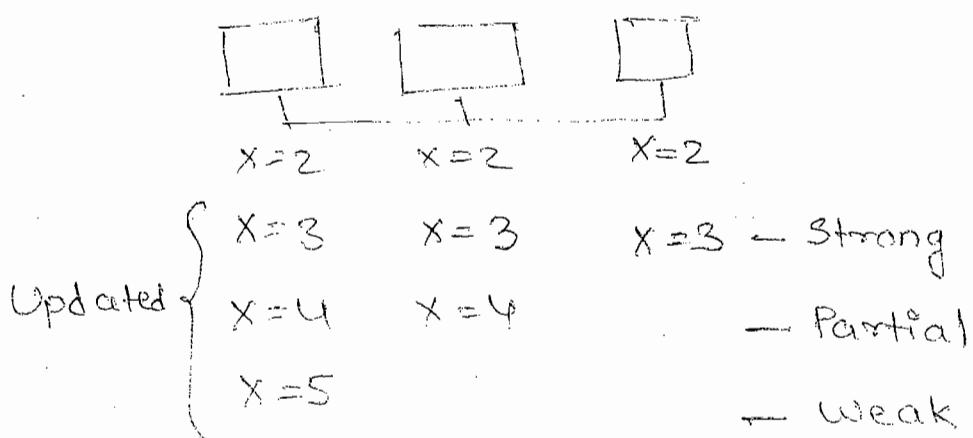
CA : Consistency Availability \rightarrow RDBMS

AP : Availability & Partitioning \rightarrow Cassandra

CP : Consistency & Partitioning \rightarrow HBase

Consistency \rightarrow divides into three parts -

- Strong consistency
- Partial consistency
- Weak consistency



Nosql Databases — categories in 4 types —

① Key-Value databases —

eg :- $\text{emp}(25) = \text{"praveen"}$

$\text{emp}(26) = \text{"Huma"}$

Types -

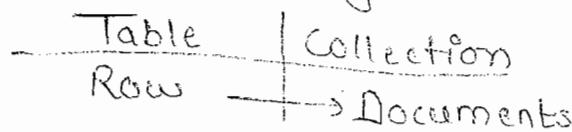
↳ Raik, Dynamo DB, Accurate

\uparrow
Nasa
uses
Raik

③ Document Based Database :-

Ex → EMP DEPT PERSO OFFON

- joint opern is very costly in SQL
- collectn of key value record.
- No structure defined
- Row is nothing but Document



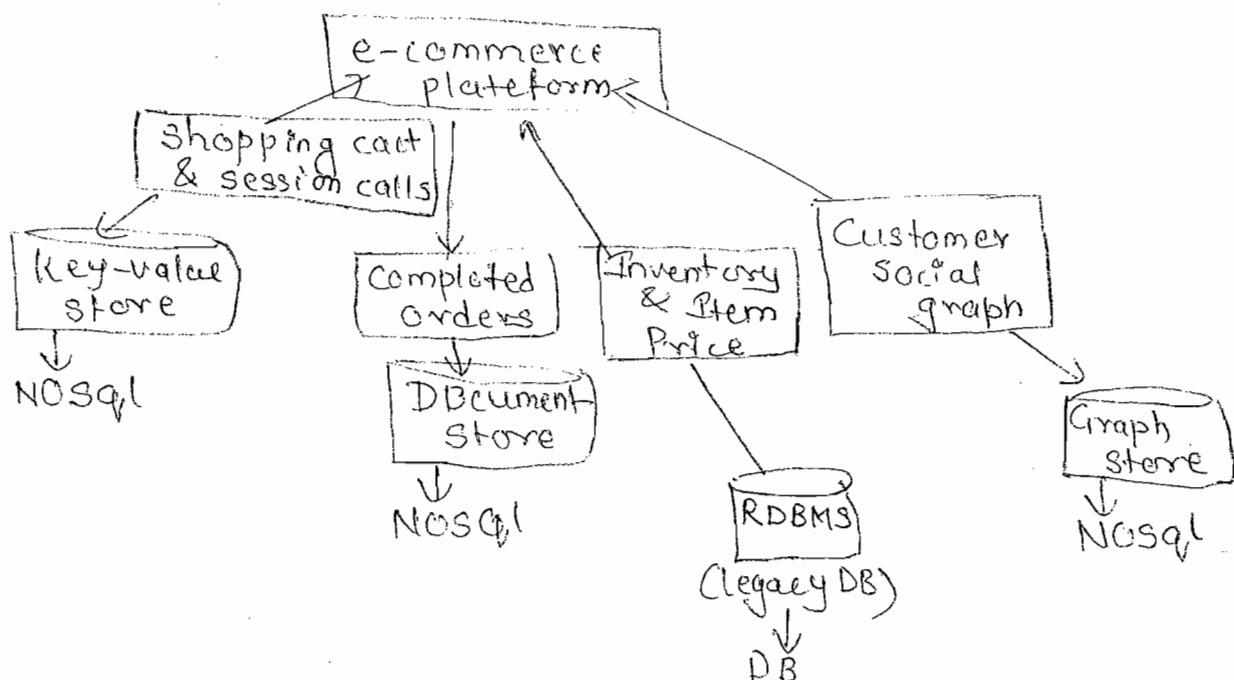
Eg → MongoDB, CouchDB and ArangoDB

④ Graph-oriented Database :-

Ex → Three types of Graph-oriented Database -

- ① → Spatial Graph (BFS, spanning tree, ex) → Relationship b/w two different point
- ② → Network Graph (distance between two servers)
- ③ → Social Graph (Relationship b/w people is called Social graph)

Eg → Neo4j
Neo4j

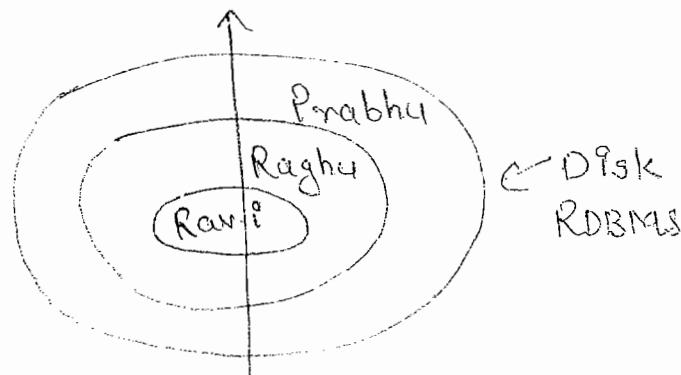


3) Column Oriented Databases -

Row
Oriented DB
(RDBMS)

emp	salary	age
Prabhu	5000	23
Raghv	6000	24
Ravi	7000	28

If we go for aggregation? Then -



In RDBMS, it scanned all the entire table first and ~~it is~~ then it is going to check for operation. It takes time & also a big disadvantage.

Column

Oriented → e.g. Prabhu | Raghv | Ravi

DB

5000	6000	7000	→ Salary
23	24	28	→ Age

Now if we go for aggregation -

Read → Row

Operation → Column

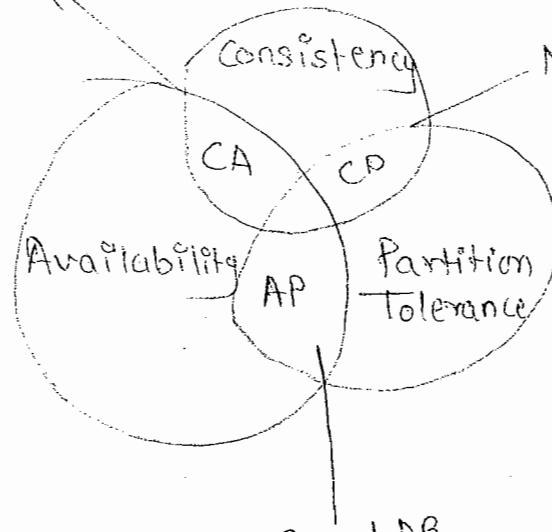
⇒ Column Oriented DB is good only for aggregation

↓
e.g. → HBase → ~~facebook~~
→ CASSANDRA

HBase →

→ HBase uses file system → HDFS.

RDBMS CAP Theorem —



MongoDB
HBase
Redis

Note:- HBase is in
Top of
Hadoop
Not part of
Hadoop

→ Totally different
from each other.

CouchDB
Cassandra
DynamoDB
Riak

RDBMS — Row oriented DB

Create Table



① Structure (schema - Metadata)

② Store Data

In Column Oriented DB is totally different —

↳ Create Table



Column-FAMILY

Column & data
will be inserted

Column
data

→ fly

Then aeroplane
is Hbase

HBase → Aeroplane
e.g. want to go
some place in a very
less time / latency
Batch data is High
latency.

Normal data is
less latency

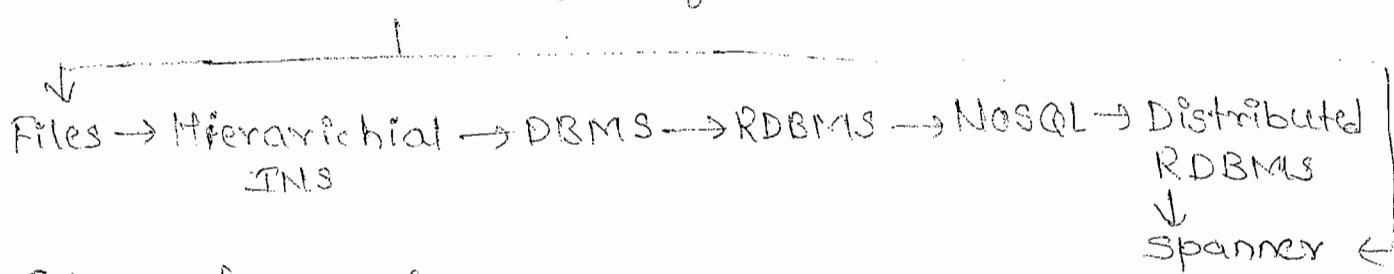
Spark will produce
result in m/m

for e.g:- we take an emp personal table

Table →

Name	address	phone no	Office-No	Office_Address	Office-phone no
------	---------	----------	-----------	----------------	-----------------

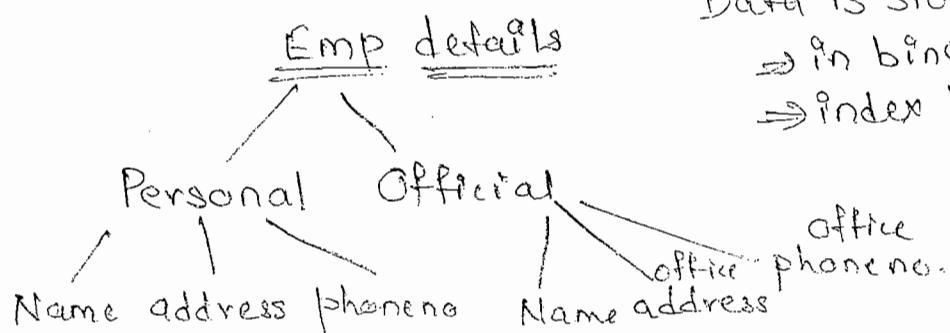
Different types of DBMS



→ Column family is nothing but subpart of your family.

→ Cf1 → Personal → Name address phoneno.

Cf2 → Official → office-No office add office-phone no.



Data is stored in HDFS
→ in binary format
→ index hdfs file

Rowkey	Timestamp	Name family			Address	family Street
		First	Middle	Last		
Row 1	T1	Mukund	I	Patel	20	Park street
	TS				10	—
	TG					
Row 2	T9	Ranjit			44	MG Road
	T12					
	T13					

11/07/2015
Saturday

→ HBase is copy from google Bigdata - GFS

↓
HDFS

Complexity
(K,v) key value
Document
Column Oriented → HBASE
Graph Oriented

	Name	Age	Office name
→ Ramesh	19	NULL	
→ Raghu	35	INFOSYS	
→ Ravi	25	TCS	

OFFICENAME OFFICE_NO

NULL

INFOSYS

TCS

SQL

Table structure (columns)
DATA

Column
Table
columns
Family
Columns
↓
Data

	Wife name	Child name
	NULL	-
SARITHA	HARI	
	NULL	-

→ Table will divide
into 2 parts
personal & official
info

Create 'emp', 'personal', 'official'

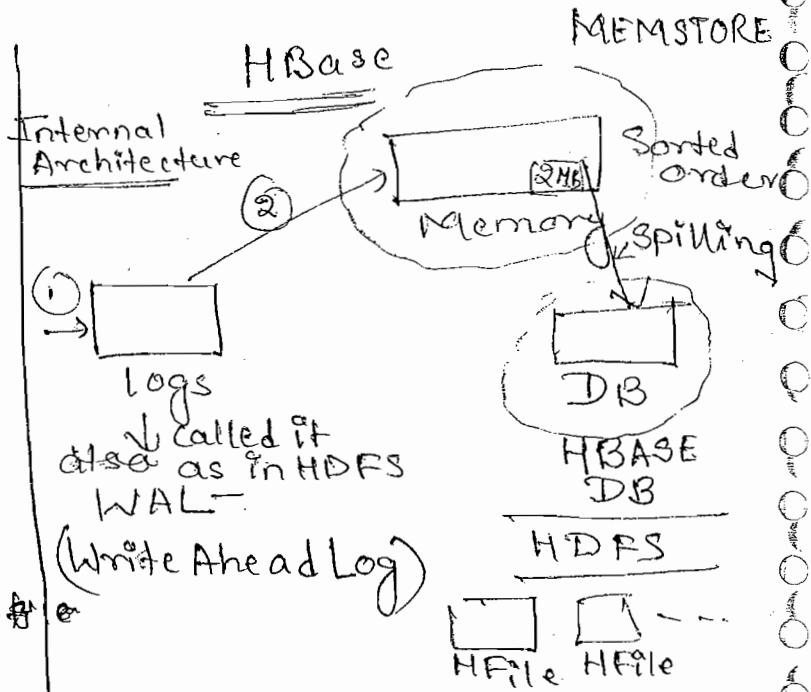
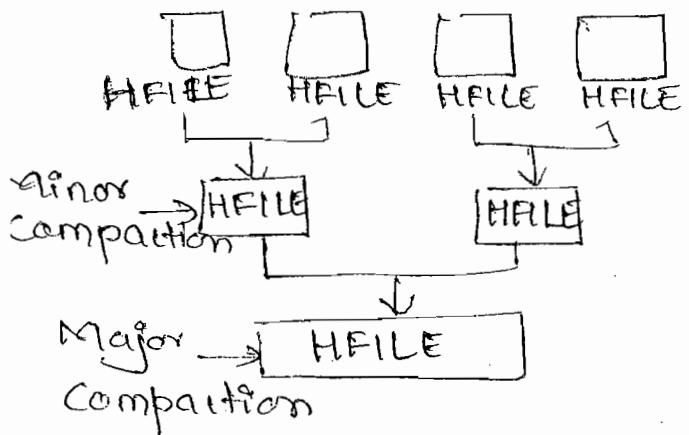
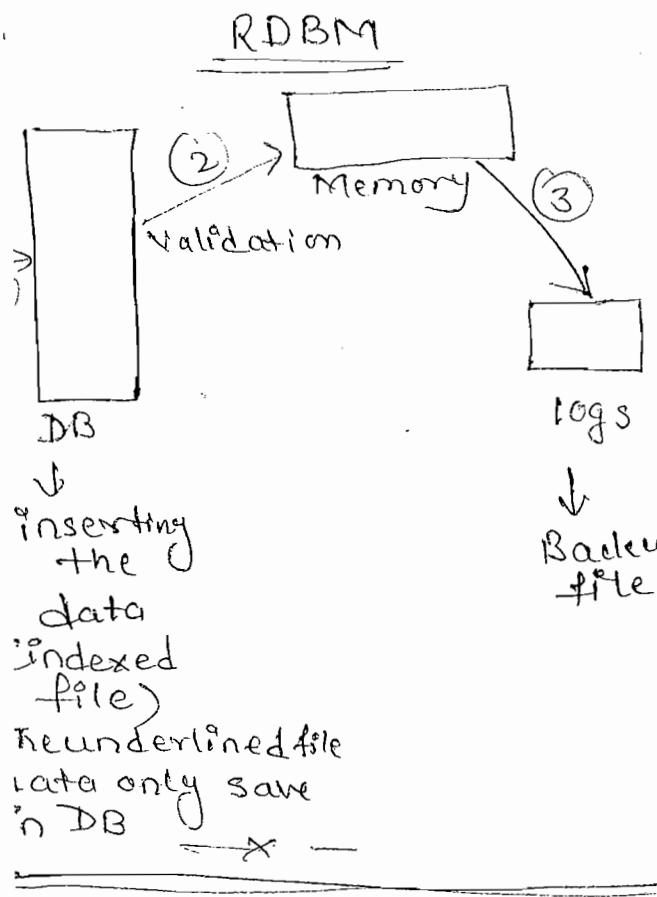
↓
Table ↓
 CF1 ↓
 CF2

Note
→ HBase does not
have null
values

Table
CF1 CF2
↓ ↓
Column calls Col Col
↓ ↓
Data Data
↓ ↓
Data

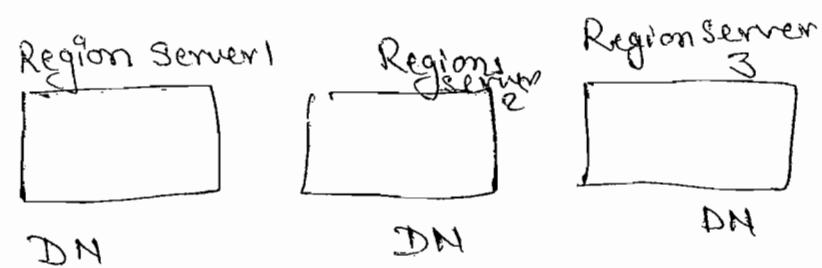
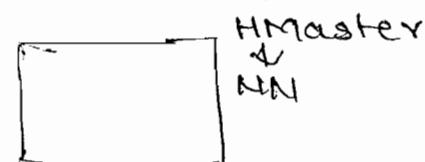
	Name	Age	Office	Personal
Ramesh	19	INFOSYS	Wife: NULL	Age: 19
Raghu	35	TCS	Wife: SARITHA	Age: 35

→ Time Stamp is a unique record key for every records.

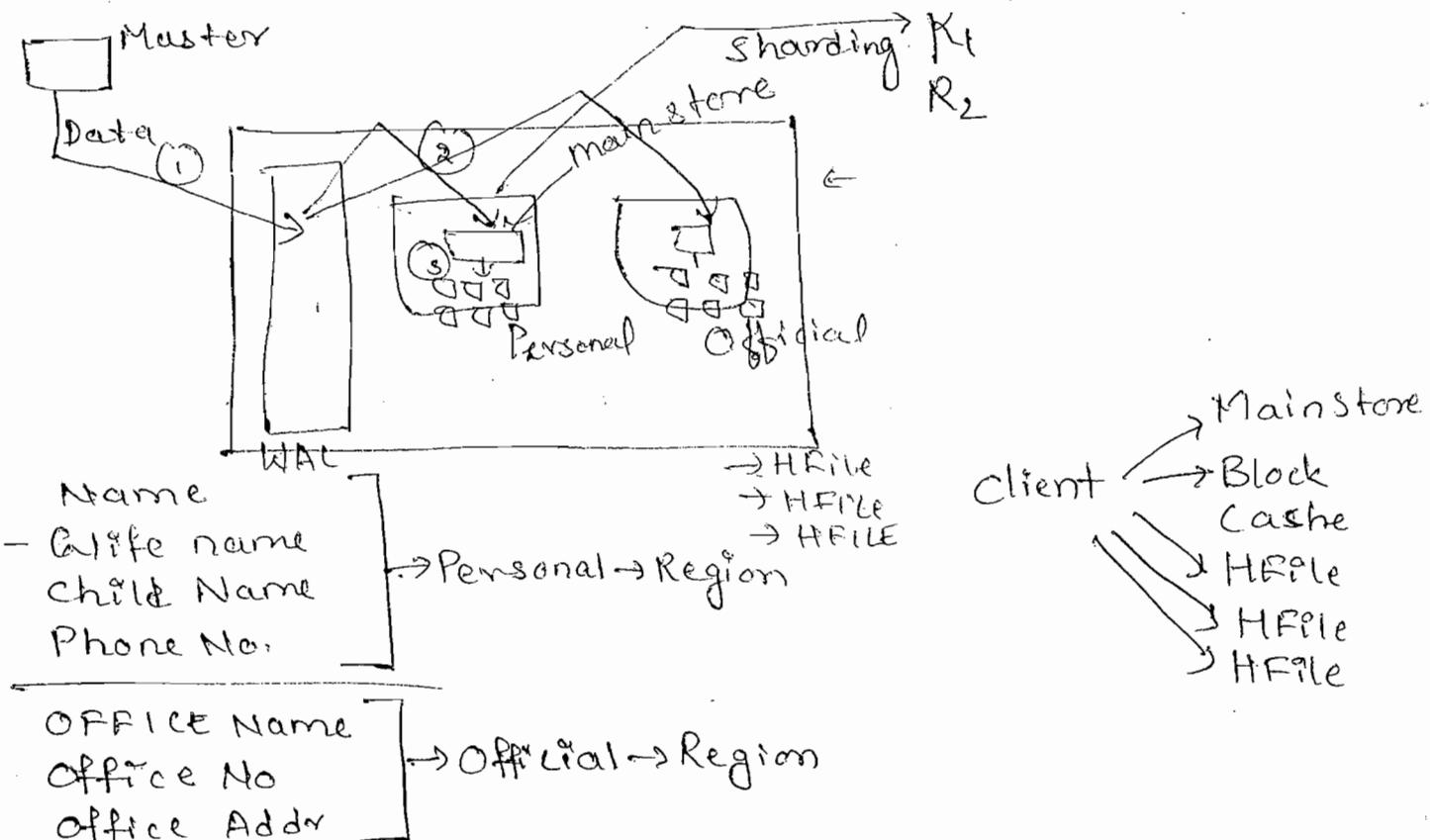


- The database stored in HBase should be in sorted order.
- Sorting will be more faster than the disk
- The underline fs of HDFS is HBase
- After certain limit the data is spill \ into files
- Hfile is Index File.
- Spilling the data in DB.
- Replication helps to restore the full HBase in their 3 nodes.
- Again, for every spill we create a new Hfile.
- Combining the HFILE is called Compaction
- Compaction has 2 types.
 - ① Minor Compaction
 - ② Major Compaction

- Minor compaction happens at regular interval and do not need to utilize more resource.
- Time taken is less by the compaction method
- Major compaction happens once at a time
- The data flow in HBase is totally diff. from the traditional flow.
- In tradⁿ system, the data writes 1st to the db & then emits to the logs for the backup purposes.
- Install HBase on the top of Hadoop...
if we install the master is called HMaster.



Slave for HBase → Region Server



→ Region is called Column family.

→ Collection of Hfiles is called Region

→ We insert the data in Region only

→ Mainstore spill as Hfile

→ The actual region will be stored in DN

→ Main store for every region

→ Inital will be for whole region

→ Each region ~~store~~^{have} m/m. ^{region}

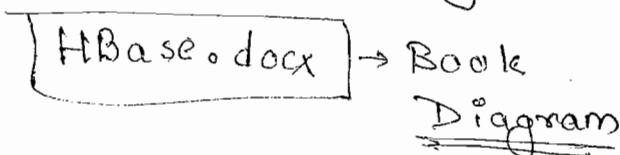
→ After certain file the m/m will split
the it. will split into region

called daughter region & the
process of splitting the region

is called sharding.

→ Splits bcoz of thresholding

→ A single column family has multiple regions.
→ But a region can ~~for~~ have only data of
^(single)
one column family.



BlockCache → The repeatedly ask data get stored
in BC.

How write will perform?
in Hfile

10/7/2015
Sunday

Start

① Hadoop → start-all.sh

② HBase

\$ bin/start-all.sh

\$ Now, start safe mode

Home installations hbase-0.94.14 & config

\$ jps

\$ cd \$HBASE_HOME ⇒ You are in HBase folder bashrc file

\$ pwd

\$ bin/start-hbase.sh

\$ jps

Goto web console

localhost:50070 — NameNode

localhost:60010 — HBase Master

Default table in hbase is catalog

① → ROOT (hyphen root)

② → Meta (dot meta)

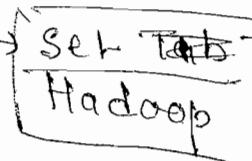
\$ bin/stop-hbase.sh

\$ jps

File

↳ Open Tabs

Vm4learning → right click →



\$ cd HADOOP_HOME

\$ rm -r log/*

\$ rm -r data/*

\$ bin/stop-all.sh

\$ bin/start-all.sh

\$ jps

\$ bin/start-hbase.sh

\$ cd \$HBASE_HOME

\$ bin/start-hbase.sh

Go to Browser

HBase Master

Open → The region Server —

Vm4learning, 60020

→ Like any query, HBase provide a shell called HBase Shell.

→ We use shell for debugging purpose (in real time)

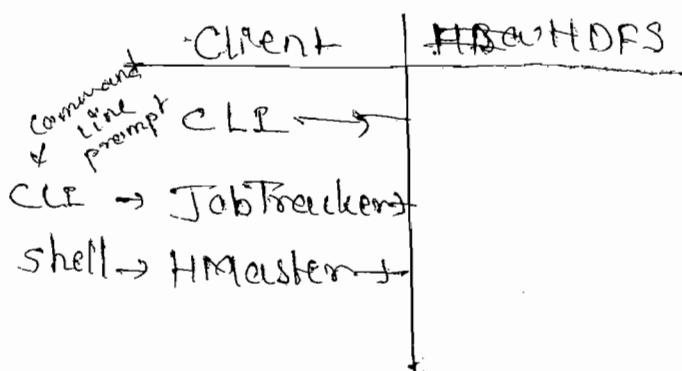
\$ jps

\$ bin/hbase shell → it login you to HBase shell.

→ This HBase shell was totally written in Ruby.

hbase(main):001:0> list
:002:0>

Show tables list



Goto Hbase-site.xml - /Installation/hBase
<configuration>
<property>

<name>hbase.rootdir</name>
<value>hdfs://

Goto Browser

HDFS! /hbase

Goto: /hbase → create for every split
→ nothing but a binary file where we can see the data
→ no datatypes in hbase
→ default datatype of hbase is byte array

Creating Table 'employee' 002:0 > Create 'employee', 'personal'
003:0 > list
004:0 > scan 'employee'

Show tables | list
Select * from | scan

Got a NameNode
HDFS/hbase

Got a /hbase/employee/a5f2-----ah

Region

Got a HBase Master : vm4learning

→ employee | Online Register | Description
UserTable | ↓ | 'employee', {Name
↓
a5f2923760

Now, we INSERTING THE DATA

emp-Id → Row key → row key can be duplicate
employee → Task

personal → column family

name → Column Qualifier

'Raghav' → Value

PUT 'tablename', 'Rowkey', 'Columnfamily'
COLUMNQUALIFIER', 'value'

Now put all the values -

put 'employee', '12345', 'personal:name', 'Raghу'

005:0> scan 'employee'

~~12345~~

006:0> describe 'employee'

007:0> scan 'employee'

008:0> put 'employee', '12345', 'personal:name',
'Raghу'

Now we want to
add age to the table

12345 Raghу 32 Hari

12345 Raghу

12345 32

12345 Hari

009:0> put 'employee', '12345', 'personal:age',
'32'

010:0> put 'employee', '12345', 'personal:wife',
'Haritha'

011:0> scan 'employee'

Goto browser

NameNode console

↳ filesystem Region name
/hbase/employee/ = - - -

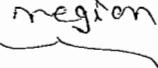
We do not find data here

To see the data we have to stop shell

[Control + Z]

\$bin/stop-hbase.sh → stop the hbase
\$jps

Browser

/hbase/employee/ /region

Now again start the hbase

\$bin/start-hbase.sh
\$jps

\$bin/hbase shell

shell is case sensitive

001:0> list

002:0> scan 'employee'

003:0> put 'employee', '123456', 'personal:name',
'Kajal'

004:0> put 'employee', '123456', 'personal:age',
'23'

005:0> put 'employee', '123456', 'personal:
phone_no', '9866312756'

006:0> scan put 'employee', '123456',
'personal:address', 'Hyd'

007:0> scan 'employee'

Control + Z stopped

\$bin/stop-hbase.sh

⇒ Again check the browser as for previous

\$bin/hbase shell

001:0> list

001:

\$ bin/start-hbase.sh → start hbase

Login to the shell

\$bin/hbase shell

001:0> list

002:0> scan 'employee'

003:0> get 'employee', 123456, 'personal-age'

Value=23 ← we got.

Now we want phone no.

004:0> get 'employee', 123456, 'phone' ← ~~why~~

By this we can
got all the information
of employee.

005:0> get 'employee', 123456, 'personal':
Phone-no

<u>DDL</u>	<u>DML</u>
→ Create	Put get

RDBMS -

→ Hides the detail of the history

example → Gives present wife name
not past

HBase -

→ It provides History too

→ It checks the time stamp

009:0> put 'employee', '12345', 'personal:
wife-name', 'Kajal'

^{wife} ← Before it is

Sanitha

010:0> get 'employee', 12345@

011:0> put 'employee', '123456', 'personal:
Phone-no', 'XXXXXXXXXX'

← Now it is Kajal

012:0> get 'employee', '123456'

013:0> scan 'employee'

Version in hbase → You can see History

014:0> describe 'employee'

VERSIONS=3

↳ Going to maintain last
3 updates

015:0> scan 'employee'

016:0> scan 'employee', { VERSIONS=>3 }

Now, we again change the wife name for 'Raghu'

018:0> put 'employee', '12345', 'personal:wife',
 Sanitha

019:0> scan 'employee', {VERSIONS=>3}

020:0> Scan 'employee', {VERSIONS=>3}

Again change the wife name for
Raghu

021:0> put 'employee', '12345', 'personal:wife',
 priyanka

022:0> scan 'employee', {VERSIONS=>3}

It will give all the 3 names of the wife
(past & present)

To remove these problem

we will set VERSIONS=>1

↳ only one wife name will
be give

022:0> delete 'employee', '12345', 'personal'.

023:0> scan 'employee'

024:0> delete 'employee', '12345', 'personal'
 age'

Again it will ^{not} ~~the~~ delete the age

It will deleted only by the API packages.

8/07/2015
SATURDAY

\$ cd \$HBASE_HOME
\$ bin/start-hbase.sh
\$ bin/hbase shell

001.0) list
002.0) scan 'employee'
003.0) describe 'employee'
004.0) create 'students',
'marks', 'affairs'
005.0) describe 'students'
006.0) put 'students', '1234567', 'affmarks',
'marks:English', '90'
007.0) put 'students', '1234567', 'affairs:name',
'Anila'
008.0) put 'students', '06H91AO545', 'marks:AP',
'98'
009.0) scan 'students'

010.0) list

011.0) alter 'employee', {NAME=>'official', VERSION=5}
012.0) is-enabled 'employee'

→ true ↳ is the comment, whether your table
 is enable or disable.

if it give true means it is enable.
and also we cannot perform any alter operation.

Create table , cf₁, cf₂

two column family

\$ cd \$HADOOP_HOME
\$ bin/start-all.sh
\$ cd \$HBASE_HOME
\$ bin/start-hbase.sh

How to

start HBase

013.0) disable 'employee' → it is used to disable

014.0) is_disabled 'employee' the table

→ true → the table is disable

therefore it gives 'true' value

and know we can alter the table.

015.0) alter 'employee', {NAME⇒'official',

VERSION⇒5}

→ Updating all regions with new schema..

016.0) is_enabled 'employee'

→ false

Now, we can enabling the 'employee' table again.

flow chart

disable



Alter



Describe



enable

017.0) enable 'employee'

018.0) scan 'employee'

019.0) describe 'employee'

020.0) put 'employee', '123456', 'official:lname',
'Infosis'

021.0) disable 'employee'

022.0) alter 'employee', {NAME \Rightarrow 'official',
METHOD \Rightarrow 'delete'}

This method is used to dropping the column family...

023.0) describe 'employee'

VERSION \rightarrow gives the history of updates

TTL \rightarrow when this column family deleted automatically & also it is unlimited.

BLOOMFILTER \rightarrow it is useful in 'reads only'.

- If the record already exists then go for checking the existing record.

Put '450'

hash(key)%8

hash(450)%8 \Rightarrow 2

hash(500)%8 \Rightarrow 5

this kind of situation we called as 'True Negative'

already 6 is there

hash(1000)%8 = 6

for that the Bloomfilter

is flop for that in

real time they take 1024x1024 bits, Bloomfilters are useful in not a small 8 bits... 'Read' only

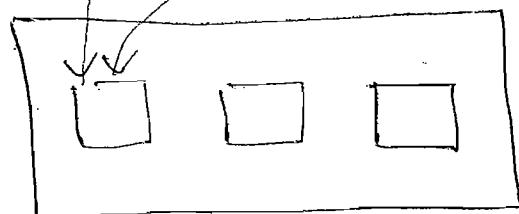
1	1	1	1	1	1	1	1
0	1	2	3	4	5	6	7 bits

get 450

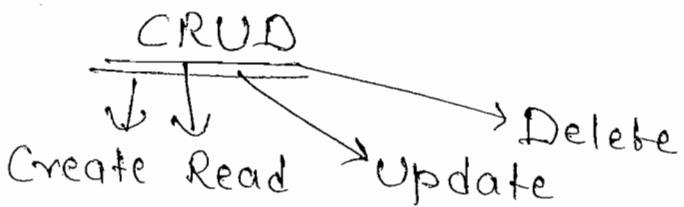
- hash(450)%8
 \rightarrow 2

hash(900)%8 = 4

If already exists then go for records.



Region Servers



Info :-

Project Explorer

↳ HBase-CRUD

↳ src

↳ default-package

↳ CreateTable.java → RUN

↳ PutExample.java → RUN

Info
Hbase

001.0) list

002.0) create 'mytablef', 'cf1'

003.0) scan 'mytablef'

004.0)

BIG Difference
between both

Now open

↳ PutExample.java → RUN

004.0) scan 'mytablef'

Now open

↳ GetExample.java → RUN

005.0) get 'mytablef', '123456', 'cf1:col1'

Now → Value - 123456
we get the

Now open

↳ MyScan.java → RUN

```
$ bin/hadoop jar /home/vm4learning/Desktop/  
HBASE-CRUD.jar MyScan
```

```
$ bin/hadoop jar /home/vm4learning/Desktop/  
HBASE-CRUD.jar - - - - -
```

Now Open

↳ MyDelete.java → RUN

```
006:0> Scan 'mytable'
```

Now Open

↳ ListTables.java → RUN

```
007:0> Scan 'mytable'
```

Now Open

↳ DropTable.java

↳ TableExists.java

②nd ↳ AddColumn.java → RUN

①st ↳ DisableTable.java → RUN

Note -

If you want to do
any DDL operation,
do disable the table
first except 'create'
operation.

```
008:0> Scan 'mytable'
```

```
009:0> describe 'emp'
```

```
010:0> Scan 'emp'
```

↳ ListTables.java → RUN

↳ EnableTable.java → RUN

Goto the

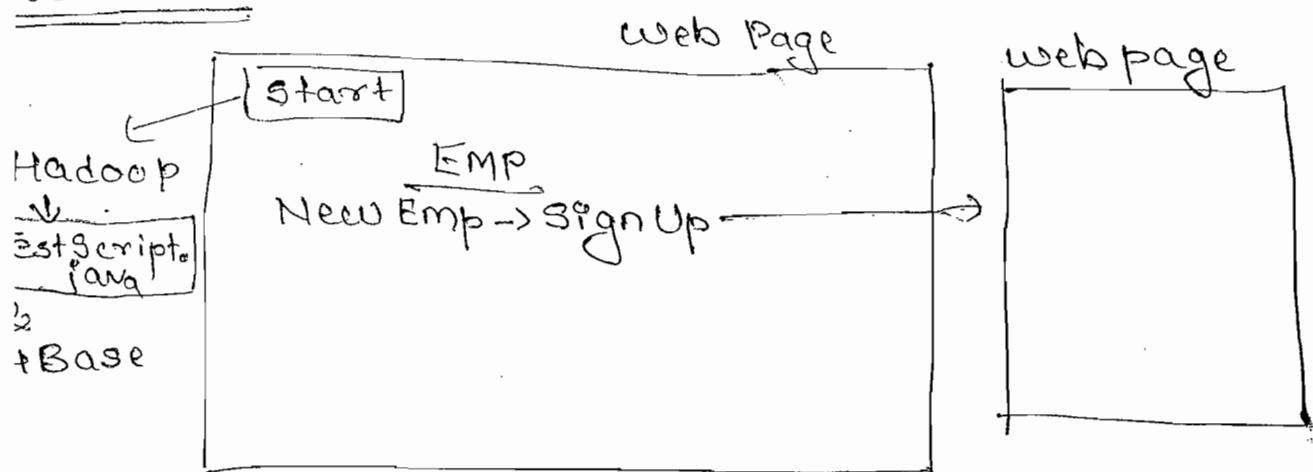
↳ HBaseTest.java → RUN

Goto

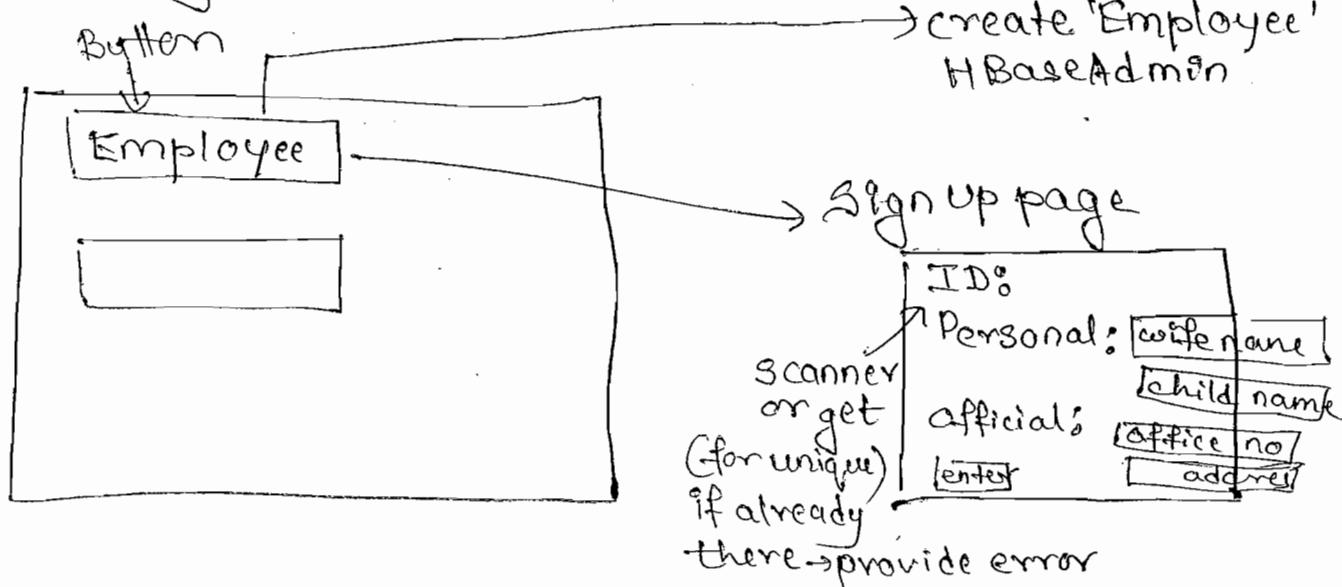
↳ ShutDownHBase → RUN

\$ jps → your HRegionServer will shutdown

PROJECT



TestScript.java ⇒ bin/start-all.sh



GET DETAILS

Emp-id :	<input type="text"/>
Name	<input type="text"/>
Add	<input type="text"/>
Wife name	<input type="text"/>
<input type="button" value="update"/> <input type="button" value="Submit"/>	

If all records are already exists
then go for sign in Button

Note

Do it with the help of Ajax...

