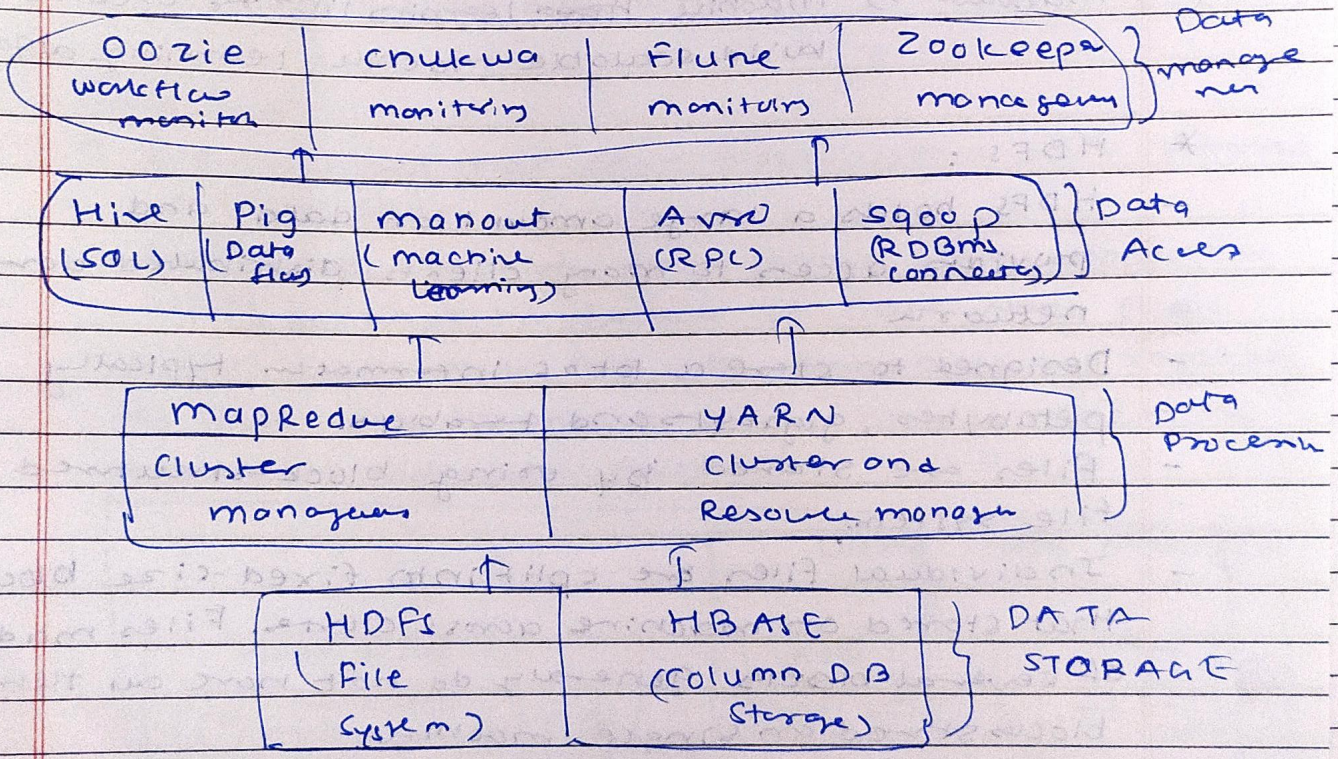
Unit 4

* Hadoop ecosystem:
  - Apache Hadoop ⇒ open source framework to make interaction with big data easier.
  - Hadoop was run large data sets which are sensitive and needs efficient handling.
  - It enables processing of large data sets which resides in form of clusters.
  - Hadoop ecosystem is platform which provides various services to some big data problems.
  - Four major elements is. HDFS, mapReduce, YARN and Hadoop common.
  - All the tools work collectively to provide services such as absorption, analysis, storage and maintenance of data, etc.

HADOOP ECOSYSTEM



(1) HDFS → primary storage system of Hadoop and is used to store large files.

(2) YARN → cluster management technology helps in managing resources and scheduling tasks.

(3) Map Reduce → programming model used for processing large data sets.

(4) Hive and Pig → query-based processing tools uses to write SQL like queries to analyze data.

(5) HBase → NoSQL database provide real-time read write access to large datasets

(6) Oozie → workflow scheduler system used to manage Hadoop jobs

(7) Sqoop → It is tool used to transfer data bet" Hadoop and relational database.

(8) zookeeper → centralized service to maintain configuration information

(9) Mahout → machine learning library used to build scalable machine learning algo's.

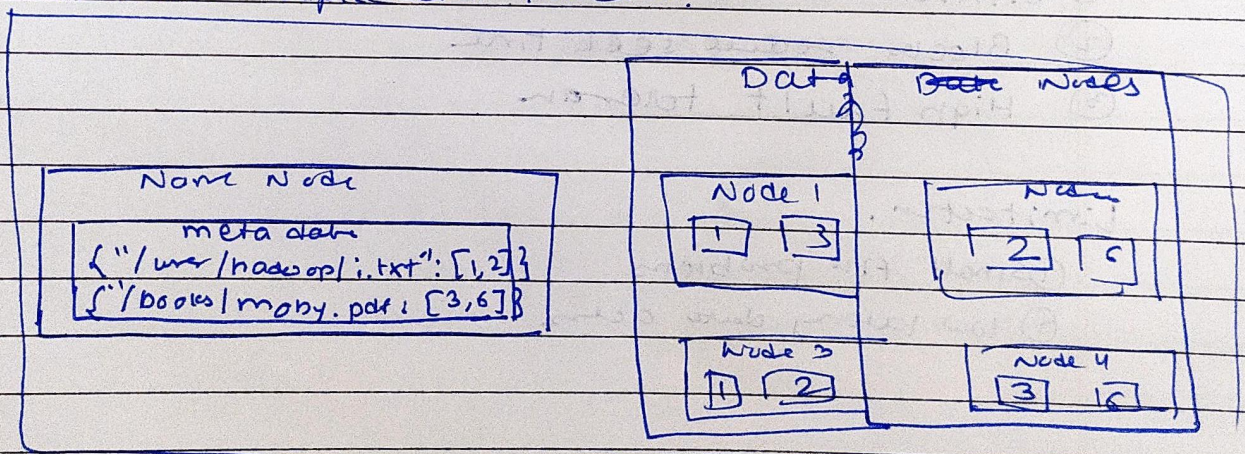
* HDFS :

- HDFS holds a large amount of data and provides access to many clients distributed across network
- Designed to store a lot of information, typically petabytes, gigabytes and terabytes.
- Files are stored by using block-structured file system.
- Individual files are split into fixed-size blocks that stored on machine across cluster. Files made of several blocks generally do not have all their blocks stored on single machine.
- HDFS ensures reliability by replicating blocks and distributing replicas across cluster. Default replication factor = 3, each block exists three times on cluster.

- Architectural design of HDFS composed of two processes:
  (1) NameNode (2) DataNode.

- NameNode: - imp. machine in HDFS.
  - It stores metadata for entire filesystem: filename, file permissions, and location of each block of each file.
  - To allow fast access, entire metadata stored in memory.
  - It also tracks the replication factor of blocks, machine failure do not result in data loss.
  - To ensure that one NameNode failure, the secondary namenode can be used to generate memory structures, reducing risk of data loss.

- Data Node:
  - The machines that store blocks within HDFS referred as Data Node.
  - Data Nodes machines have large storage capacities.
  - HDFS continue to operate if Data Node fails.
  - When Data Node fails, the NameNode will replicate the lost blocks to ensure each block meets the minimum replication factor.



HDFS cluster with replication factor 2
NameNode → mapping of files to blocks
Data Node → store blocks and their replicas.

Example: ① 100 TB data inserted, NameNode divide the file into blocks of 10 TB.

② These blocks are stored across different datanodes

③ Datanode replicate the blocks among themselves and info of blocks is sent to Namenode.

Why Divide ?

⇒ Difficult to store 100 TB file on single machine

⇒ Read and write operation will take lot of time.

⇒ multiple blocks of 128 MB become easy to read and write operation

- Terms:

① Heartbeat : signal sent by datanode to Namenode continuously. If namenode don't receive heartbeat it will consider Datanode is dead.

② Balancing : Datanode is crashed, master node will give signal to datanodes to replicate the blocks of lost blocks.

Features

① Distributed data storage

② Blocks reduce seek time

③ High fault tolerance

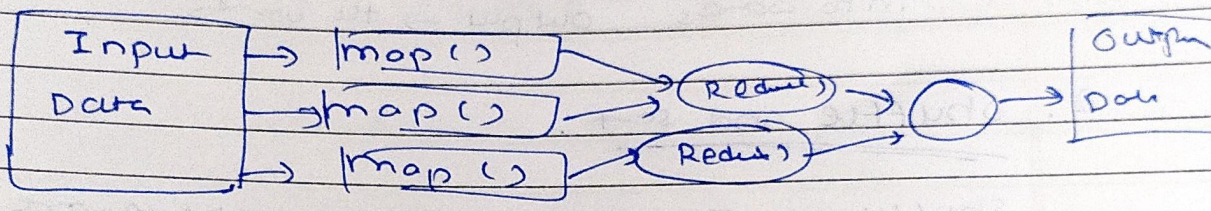
Limitation.

① Small file problems

② low latency data access

* MapReduce:

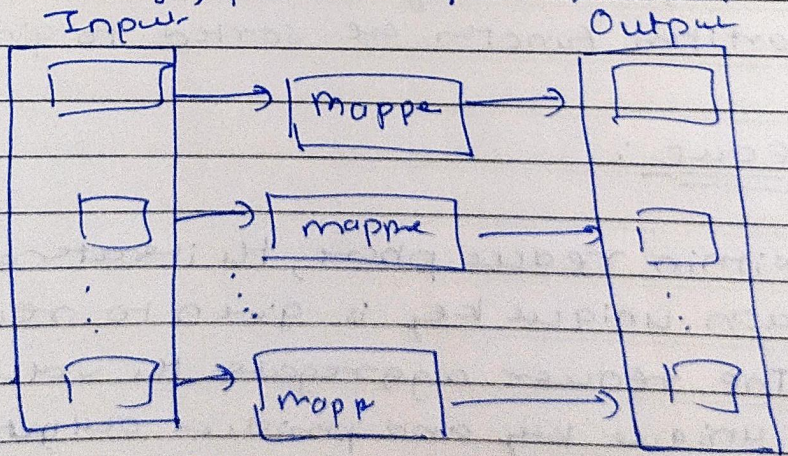
- It is a programming model that enables large volumes of data to be processed and generated by dividing work into independent tasks.
- This tasks are executed in parallel across clusters of machines.
- Every Map-Reduce program has/runs a list of input elements into list of output elements twice, once in map phase and once in reduce phase.



- MapReduce framework composed of three major phase: (1) map (2) Shuffle and sort (3) Reduce

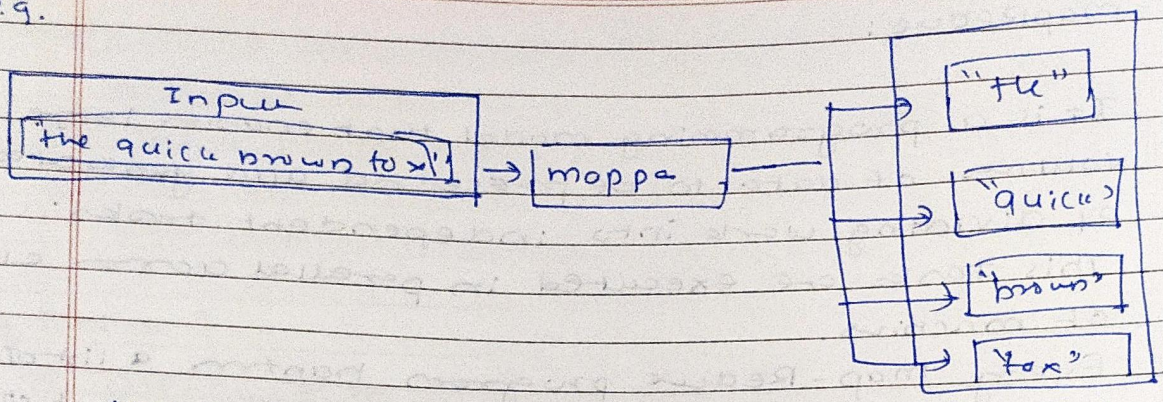
(1) map:

- Within map phase, function process a series of key-value pairs.
- The mapper sequentially process each key-value pair individually, producing output key-value pair



Mapper applied to each key-value pair producing an output key-value pair

e.g.



Above mapper transforms sentence into words.
input → sentence, mapper function → split sentence into words, output → the words.

2. Shuffle and sort:

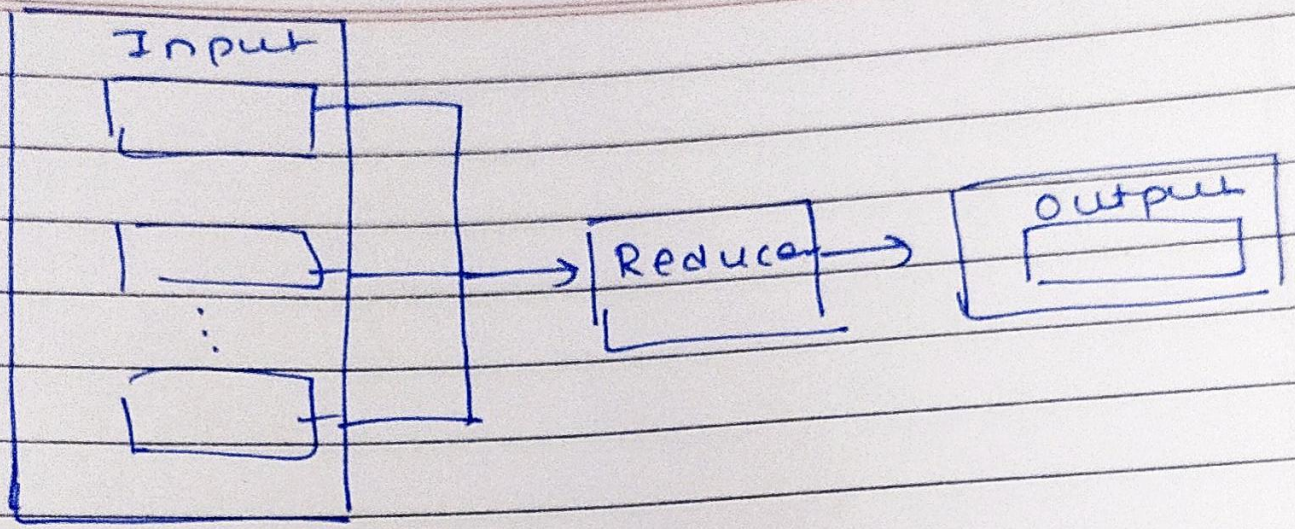
Shuffling: The outputs of map phase are moved to reducers. This process is called shuffling.

Partition function: It is used to control the flow of key-value pairs from mappers to reducers. It ensures all values of same key are sent to same reducer.

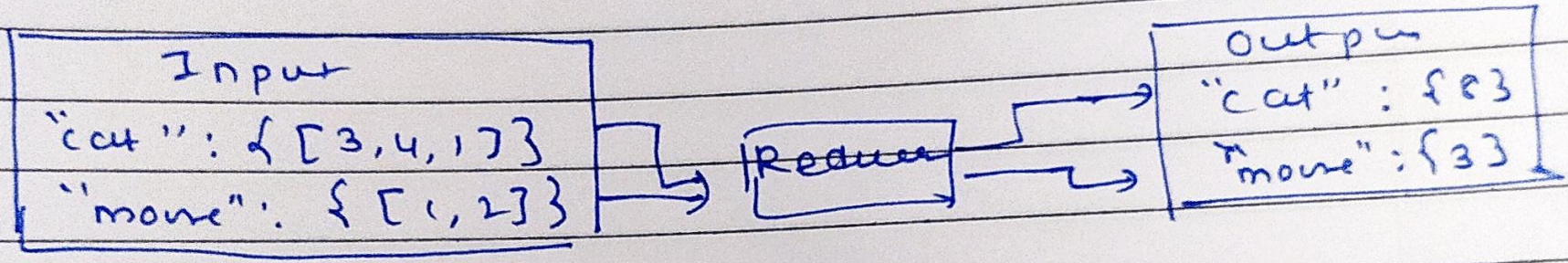
Sorting: The key and value pairs come from partition function are sorted to given to reducer.

3. Reduce:

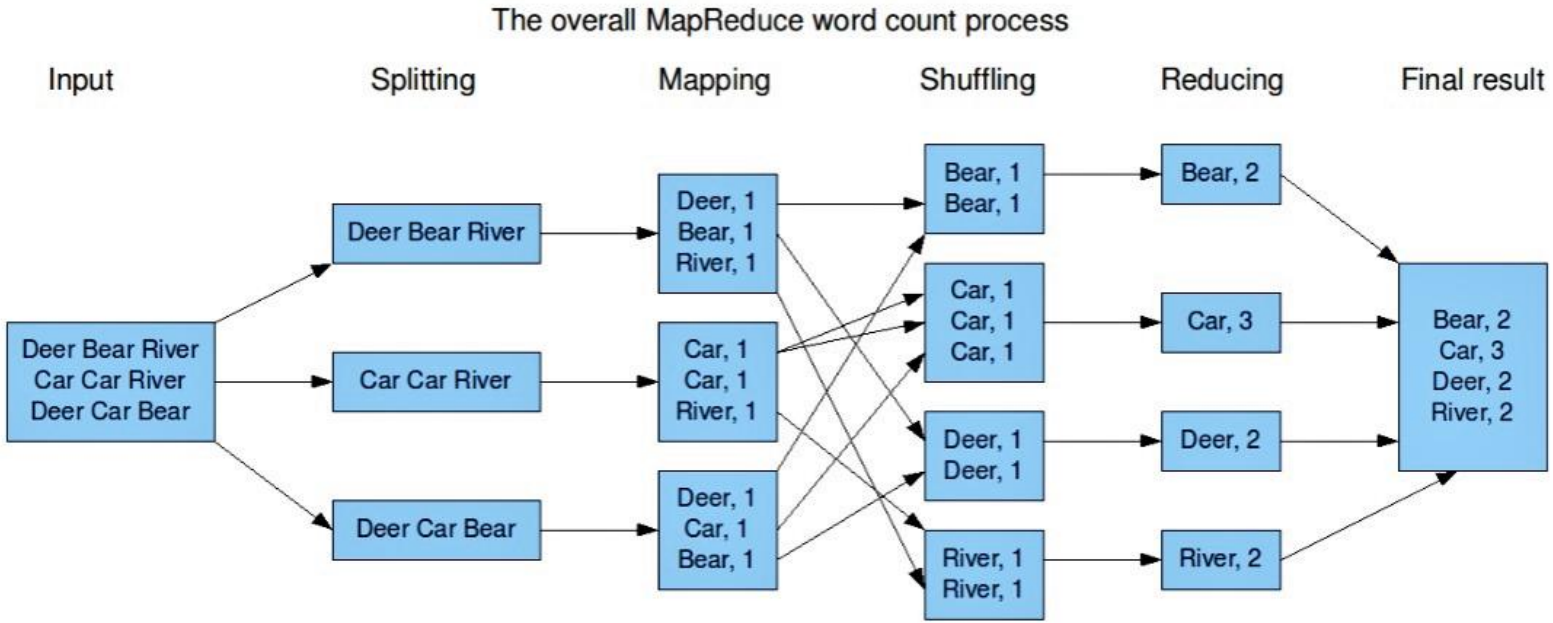
- within reduce phase, the iterator of values from each unique key is given to reducer
- The reducer aggregates the value of each unique key and produces output key-value pairs



e.g.



g. Image of mapReduce word count process by Chitty


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2