

* Hadoop :-

- Open Source software programming framework for storing a large amount of data & performing the computation.
- Its framework is based on Java programming with some native ~~C/C++~~ code.
- It is used for some adv. level of analytics, which includes ML & Data mining.
- Most well known tech. used for Big Data.
- It is a large scale ~~real time~~ batch data processing system.

* Need for Hadoop:-

- Distributed clusture system.
- Platform for massive scalable applications
- Enables parallel data processing.
- Programmers no need to worry about → where file is located, How to handle failure & data loss, How to divide computation, How to program for scaling.

* Advantages)

- Ability to store a large amount of data.
- High flexibility
- Cost effective
- High computation power
- Tasks are independent
- Linear Scaling

* Disadvantages:-

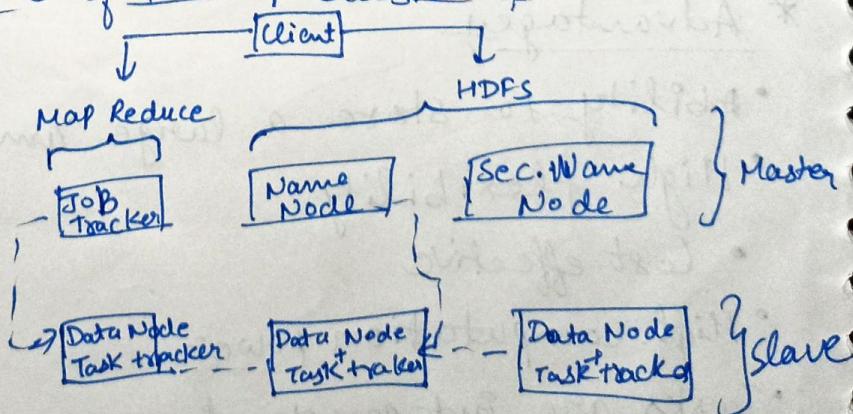
- Not very effective for small data.
- Hard cluster management
- Has stability issues
- Security concerns

* Hadoop Architecture :-

- ① Hadoop Common :- These are Java libraries & utilities required by other hadoop modules. These libraries provides file system & OS level abstraction & contains the necessary Java files & scripts to start Hadoop.
- ② Hadoop YARN :- This is a framework for Job scheduling & cluster resource management
- ③ HDFS :- A distributed file system that provides high-throughput access to application data.
- ④ Hadoop MapReduce :- This is YARN-based system for parallel processing of large data sets.

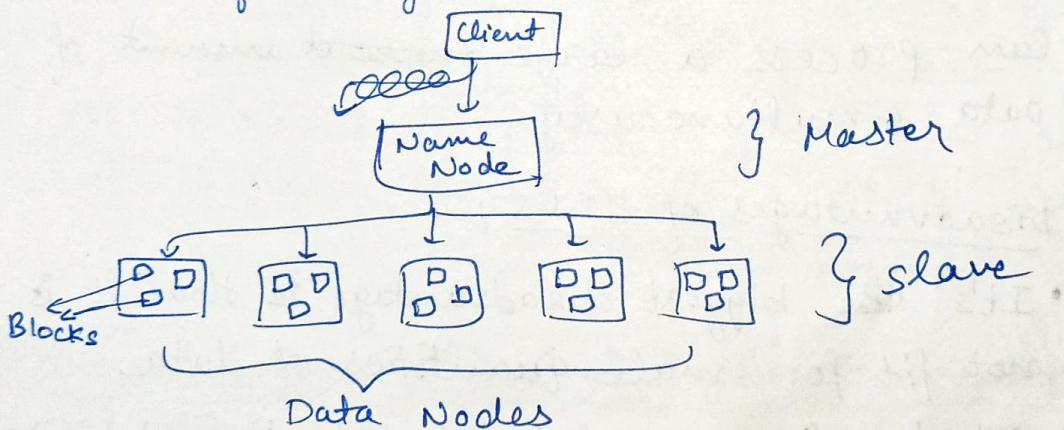
* Core components of Hadoop Cluster :-

- ① Client
- ② Master
- ③ Slave



* HDFS :-

- Hadoop Distributed File System
- A scalable, fault tolerant, High performance distributed file system.
- Asynchronous replication
- Write-once & read-many
- 3 minimum Data Nodes. cluster
- Data divided into 64 MB or 128 MB blocks.
- Each block replicated 3 times.
- ~~These~~ replicated blocks are stored in diff. Data nodes.
- HDFS consists of two core components $\xrightarrow{\text{Name Node}}$ $\xrightarrow{\text{Data Node}}$.
- HDFS maintains all the coordination b/w the clusters & hardware, thus working at the heart of the system.



* Name Node:- It is the prime node which contains meta data like,
list of files , list of blocks in each file ,
List of Data Nodes for each block , creation time

* Data Node \Rightarrow

* Data Node:-

- Stores the actual data.
- Block server stores data in the local file system.
- Periodically sends a report of all existing blocks to the name node.
- Periodic validation of checksum.
- It serves read, write request, performs block replication, deletion & creation upon instruction from Name node.

* Advantages of HDFS :-

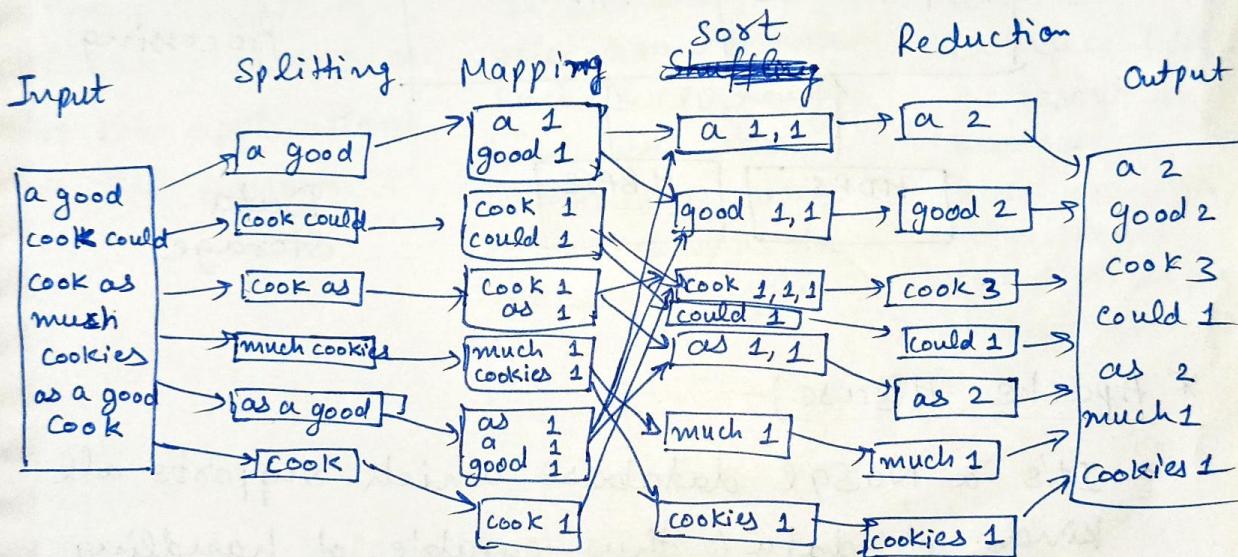
- It is inexpensive, immutable in nature, stores data reliably, ability to tolerate faults, Scalable, block structured.
- Can process a large ~~amount~~ amount of data simultaneously.

* Disadvantages of HDFS :-

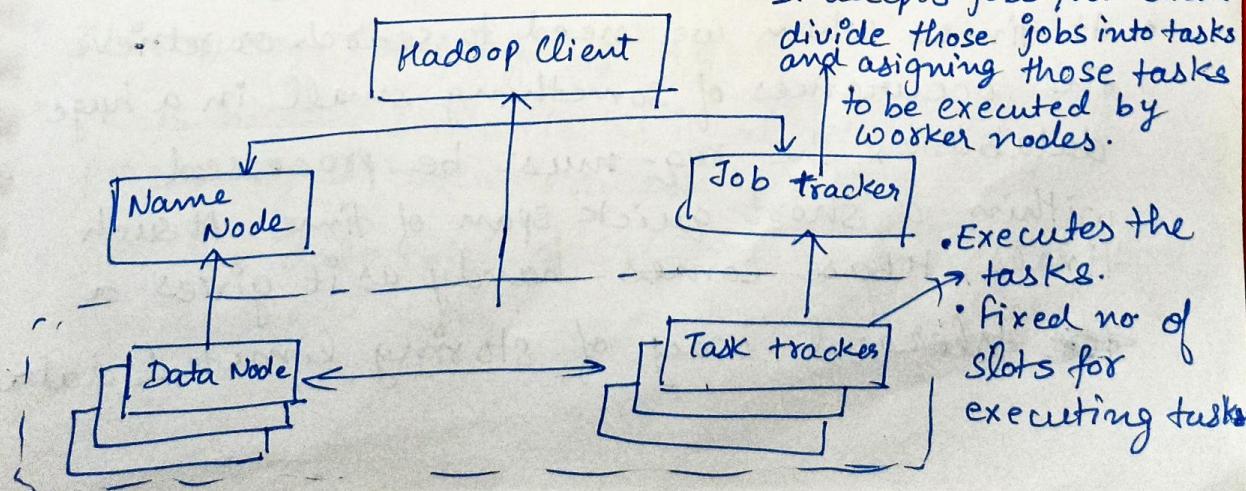
- It's ~~the~~ biggest disadvantage is that it is not fit for small quantities of data.
- It has issues related to potential stability, restrictive & rough in nature.

Map Reduce

- Software framework for distributed computation
- Input → Map() → Copy/Sort → Reduce() → Output
- Core Components of Map reduce are
 - Job tracker → schedules & manages Jobs
 - Task Tracker → executes individual map() & reduce()
- It performs the processing of large data sets in a distributed & parallel manner.



Some Common Frameworks of Hadoop



Big Data Analysis:-

- It is a process to extract meaningful insight from big such as hidden patterns, unknown correlations, market trends & customer preferences.

* Types of Big Data Analysis:-

(i) Structured Data:-

- Refers to data that has proper structure.
- for ex. → Data that is present within database, CSV files, the excel spreadsheets.

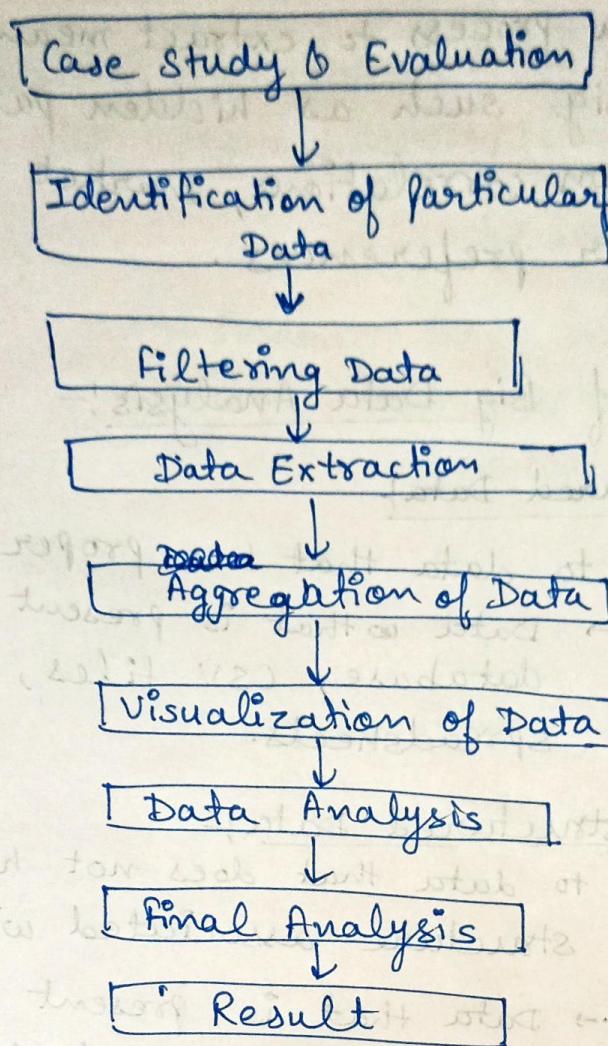
(ii) Semi-structured Data:-

- Refers to data that does not have a proper structure associated with it
- for ex. → Data that is present within emails, the log files, word documents.

(iii) Unstructured Data:-

- Refers to data that does not have any structure associated with it.
- for ex. → Image files, audio, video files.

* Process of Big Data Analysis:-



* Tools used in Big Data Analytics:-

- MongoDB :- Used on datasets that changes frequently.
- Hadoop :- Helps in storing & analysis of big data.
- Talend :- It is a tool used for data integration & management.
- Spark :- used for real time processing & analyzing large amount of data.

* Benefits of Big Data Analytics :-

- Used for risk management
- Used to improve customer experience
- Used for product development & innovations.
- Helps in quicker & better decision making in organizations.

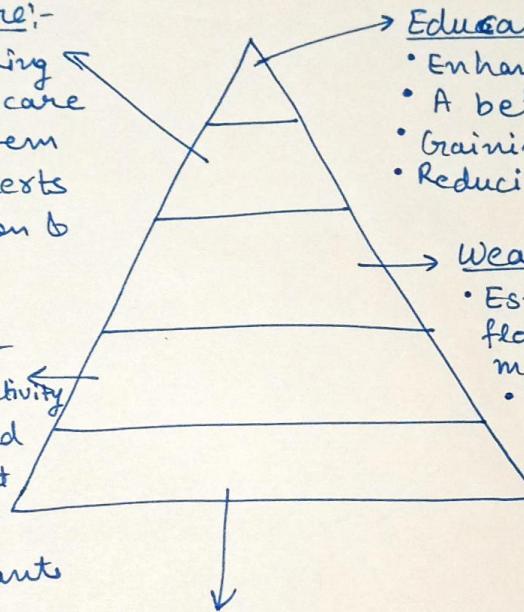
* Big Data Application Domains :-

Healthcare:-

- Health Tracking
- Improve the care delivery system
- Real time alerts
- Fraud detection & prevention.

Agriculture:-

- Boosting productivity
- Predicting yield
- Risk management
- Food safety
- Access to plants genome info.
- Savings..



Education:-

- Enhancing student results
- A better grading system
- Graining Attention
- Reducing no. of Dropouts

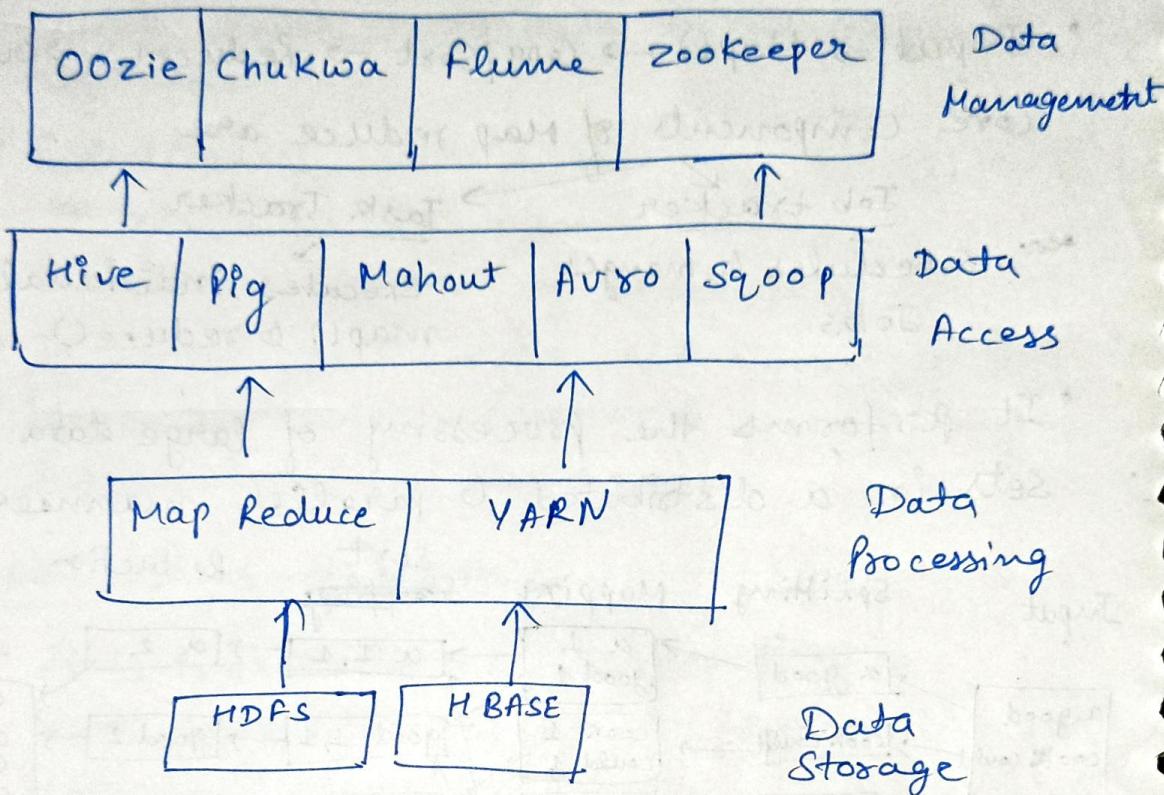
Weather Forecast:-

- Estimates of areas where flooding is likely to be most sever.
- Areas with high wind speed, snow or rain
- Locations with downed power lines
- Locations where bridges & roads most likely to be damaged

Manufacturing:-

- Product quality
- Defects tracking
- Supply management
- Manufacturing process defect tracking
- Testing & simulation of new manufacturing processes.

Hadoop Ecosystem :-



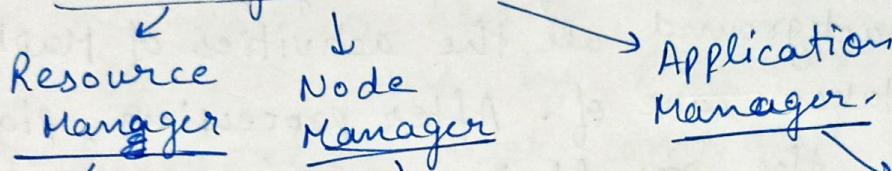
* Apache HBase -

- It's a NoSQL database which supports all kinds of data & thus capable of handling anything of Hadoop Database & hence can work with Big data set effectively.
- At times when we need to search or retrieve the occurrences of something small in a huge database, the req must be processed within a short quick span of time. At such times, HBase comes handy as it gives a ~~tolerant~~ tolerant way of storing limited data.

* YARN)

- Yet another resource Negotiator,
- It helps to mange resources across the clusters. In short , it performs scheduling & resource allocation .

- Three major components



Resource Manager
has privilege of
allocating resources
for the applications
in the system.

Node Manager
It works on the
allocation of resources
such as CPU, memory,
bandwidth per
machine & later
on acknowledges the
resource manager.

Application Manager
Works as a
interface b/w
the resource
manager &
node manager
and performs
negotiation as per
the requirement
of two.

* HIVE)

- With the help of SQL methodology, interface , HIVE performs reading & writing of large data set .
- Its query language is HQL .
- It is highly scalable as it allows real-time processing & batch processing both .
- All the datatypes of SQL are supported by HIVE hence making the query processing easier .

* PIG:-

- Developed by Yahoo which works on pig latin language & it is similar to SQL.
- Platform for structuring the data flow, processing & analyzing huge data set.
- Pig does the work of executing programs in the background, all the activities of MapReduce are taken care of. After processing, pig stores the result in HDFS.
- It runs on Pig Runtime, just like Java runs on JVM.
- It helps to achieve ease of programming & optimization & hence it is a major segment of Hadoop.

* Sqoop:-

- Sqoop is a tool designed to help users to ~~large data~~ import large data existing relational databases into their Hadoop clusters.
- Automatic Data import
- SQL to Hadoop.
- Easy to import data.
- Generates code for use in Map Reduce application
- Integration with HIVE.

* Zookeeper :-

- A high-performance coordination service for distributed application
- There was a huge issue of management of coordination & synchronization among the resources or the components of Hadoop which resulted in inconsistency, hence zookeeper overcomes all the problems by performing synchronization, inter-component based communication, grouping & maintenance.

* Avro :-

- A data serialization system that provides dynamic integration with scripting lang.

• Avro Data:-

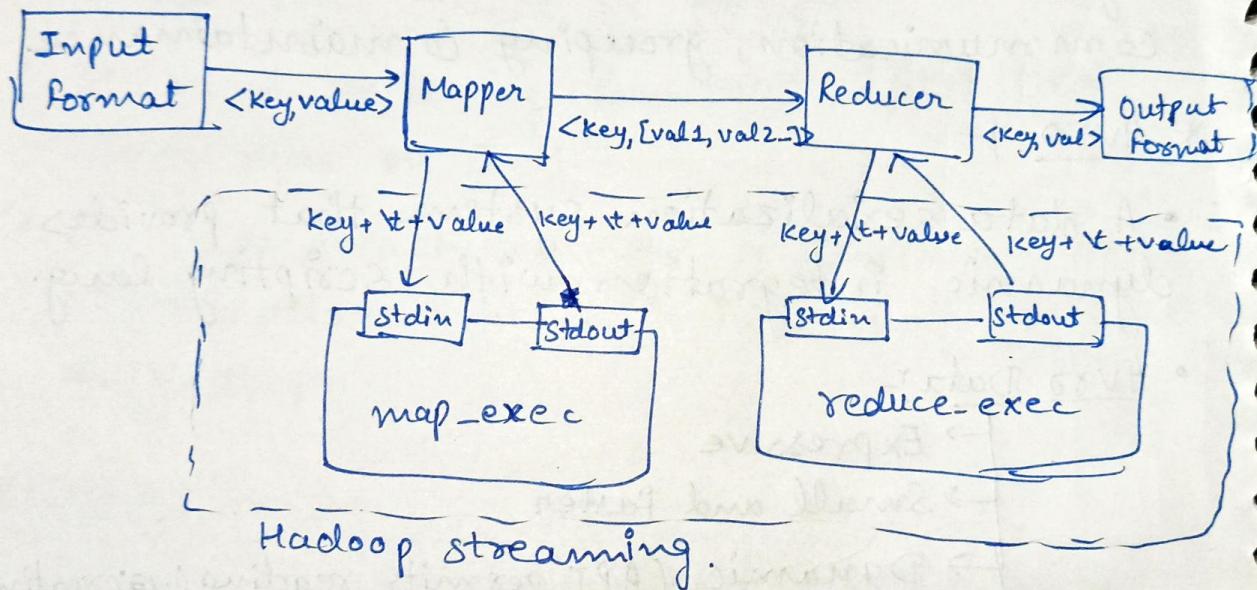
- Expressive
- Small and faster
- Dynamic (API permits reading & creating)
- Includes a file system & a textual encoding.

* Oozie :-

- It performs the task of scheduler, thus scheduling jobs & binding them together as a single unit.
- There are two kinds of Jobs i.e.
 - (i) Oozie Work flow
 - (ii) Oozie ~~Coordinator~~ Coordinator

- Oozie Workflow is the jobs that need to be executed in a sequential ordered manner
- Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

* Hadoop Pipes :-



* Hadoop Streaming :-

- Feature comes with hadoop distribution that allows developer to write map-reduce prog. in diff. languages like Python, C++, Ruby, etc.
- We can use any prog. lang. which can read from the standard input (STDIN) like Keyboard & all and write using standard output (STDOUT)

- Main advantage of Hadoop streaming utilities is that it allows Java as well as non-Java programmed MapReduce jobs to be executed over hadoop clusters.
- It translates the application logic into the ~~map~~ mapper & reduces sections with the key & value output elements.
- Three main components:- Mapper, Reducer, Driver.

* Apache Spark:-

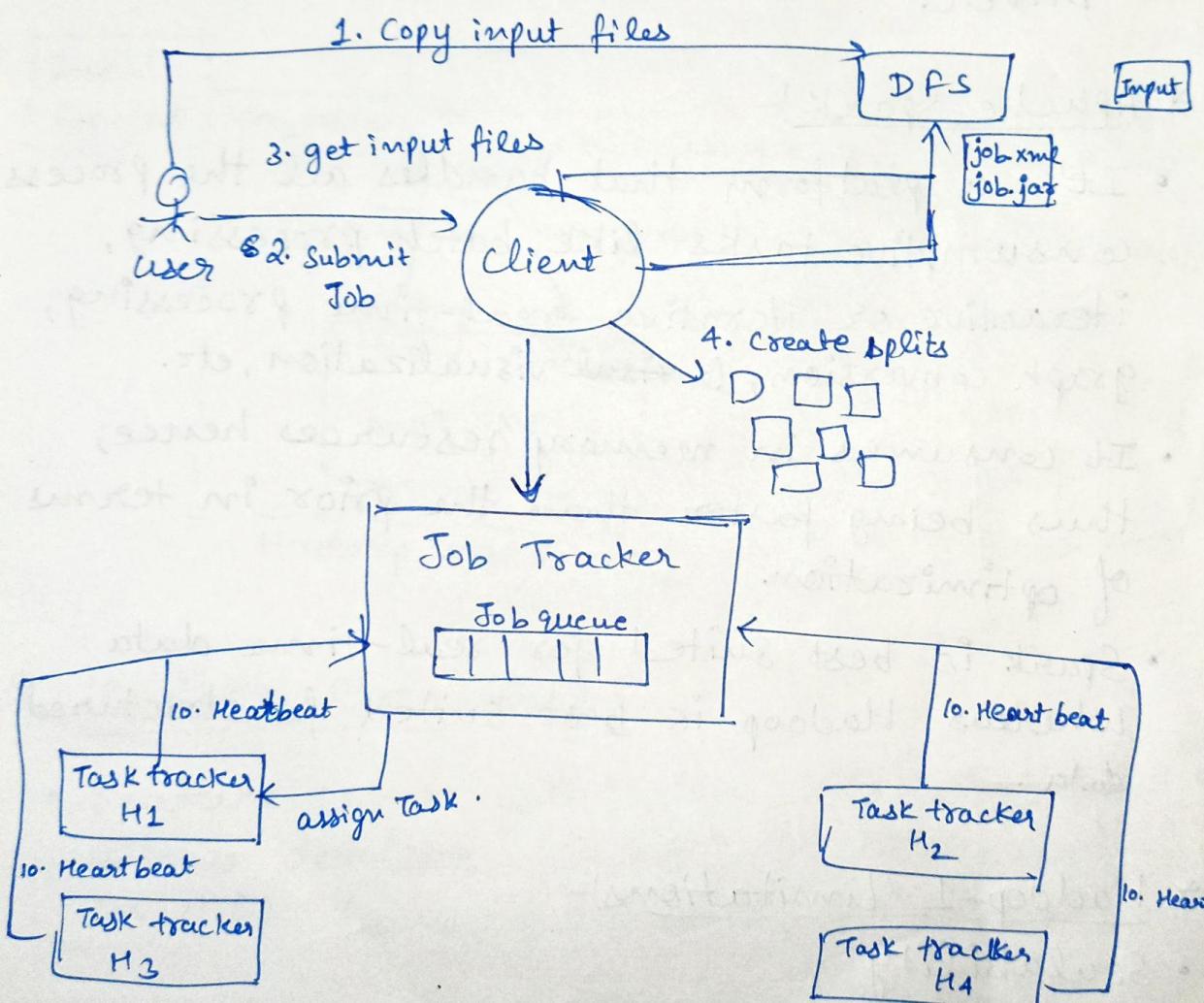
- It's a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversion, & ~~visual~~ visualization, etc.
- It consumes in memory resources hence, thus being faster than the prior in terms of optimization.
- Spark is best suited for real-time data whereas Hadoop is best suited for structured data.

* Hadoop-1 Limitations)

- Scalability
 - Max cluster size ~ 5000 nodes
 - Max concurrent tasks ~ 40,000
- Availability :- failure kills Queued & running Jobs

- Hard partition of ~~jobs~~ resources into map & reduce slots → Non-optimal Resource Utilization
- Lacks support for alternate paradigms & services → iterative application in map reduce are 10x slower.

Job tracker flow:-



Developing a Map Reduce Application :-

* Stage-1 :-

- ① Start by writing map & reduce function
- ② Then write a driver prog. to run a job, which used to run small subset of the data on your prog. to check
- ③ If it fails, use IDE debugger to find the source of problem/error.
- ④ with this, we can expand our ~~one~~ unit test and test it again

* Stage-2 :-

- ① When the prog. runs expected we can run on full dataset, it may expose some more issues, which we can fix again by above steps.
- ② Debugging failing programs in the cluster is a challenge, so we look at some ~~etc~~ common techniques to make it easier.

* Stage-3 :-

- ① After the program is working, we can do some tuning like making map reduce function faster

② Profiling distribution programs is not easy, but Hadoop has hooks to aid the process.

Anatomy of Classic MapReduce ↴

① First a job → ~~submit to~~ Hadoop ~~cluster~~ cluster.

→ generate job ID

→ validate input path

→ HDFS directory with name of job ID

② Job initialization in hadoop

→ Input splits in HDFS directory → Job Scheduler

→ Job scheduler creates one map task for ~~every~~ each input split

③ Initialized job splits into tasks and Job tracker assign tasks to task tracker

→ Heartbeat will tell status of ~~task~~ slots ~~trackers~~ i.e. are there any empty slots available

→ Now according to that job tracker will assign tasks in those slots.

④ Execution of tasks in task tracker

→ unjar contents of job jar

→ create task runner

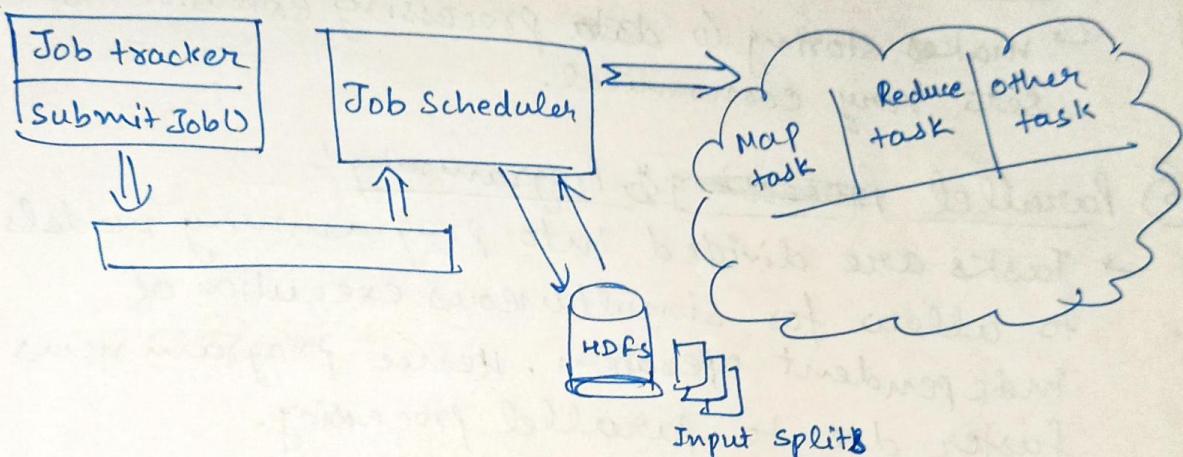
→ Launch JVM to run each task

⑤ Progress & status updates

status of job or task	progress of map & reduce	value of job • counters	status description
-----------------------	--------------------------	----------------------------	--------------------

~~Job~~ Job / task status.

- ⑥ Execution continues till all tasks completion. Once the last task completes, MR framework enters.



Key Features of Map Reduce)

① Scalability) - Map reduce can scale to vast amount of data by distributing tasks across a large no. of nodes in a cluster.

② Fault tolerance) - • ensures the reliable data processing.

- Automatically detects & handles node failure, rerunning tasks on available nodes

③ Data locality)

- takes advantage of data locality by processing data on the same node where it is stored.

hence it minimizes data movement across the network & improving overall performance.

④ Simplicity :-

- abstracts away many complexities associated with distributed computing, hence developers can focus on their data processing logics rather than low-level details.

⑤ Cost - Effective Solution :-

- ↳ makes storing & ~~data~~ processing extensive data sets very economical.

⑥ Parallel Processing & Programming :-

- Tasks are divided into programming models to allow for simultaneous execution of independent operation. Hence program runs faster due to parallel processing.