Single point of failure (SPF)

Reasons:

- Hardware failure
- Overcapacity caused by Metadata due to small data problem.

Cloudera (CDH - cloudera Distributed hadoop)

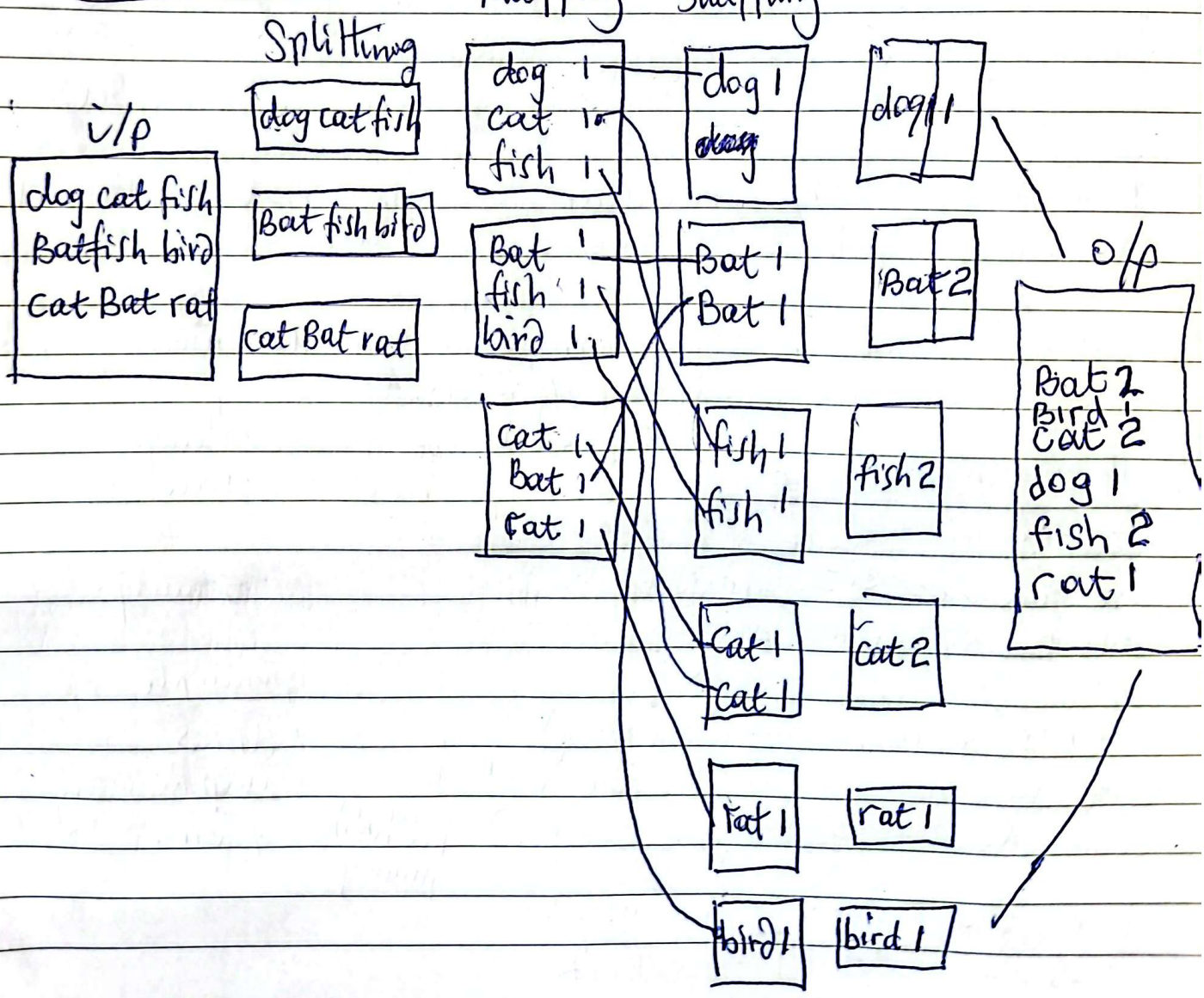
(Horton data pattern) HDP.

In 2019, Cloudera acquired +

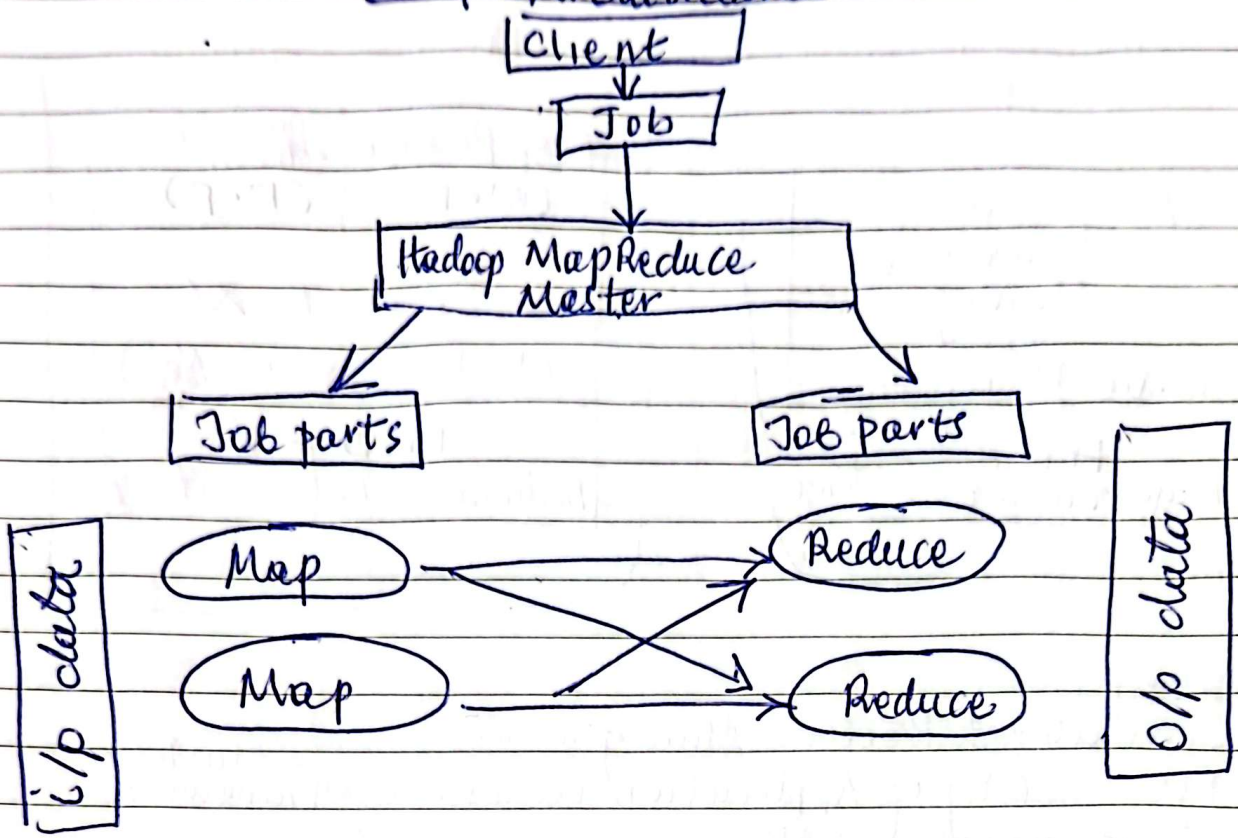
Namenode
Secondary Namenode
Datanode slave / compute / worker node ✓

Hadoop Distributed File System Hdfs

How it works



Map Architecture



YARN

Yarn means "Yet Another Resource Negotiator".
Its an important aspect of hadoop ecosystem that is responsible for decoupling / seperating the functionalities of "managing Resources" such as CPU, memories & Storage distribution from data processing as seperate components. Consequently, Global Resource Mgr can now support diverse Applications such as Spark, MapReduce, Tez, Storm etc

BA - 2012 AFTER 2012

Before 2012

Hadoop 1.0

Map Reduce
cluster Resource Management
&
Data Processing

HDFS
Redundant/Reliable Storage

After 2012.

Hadoop 2.7

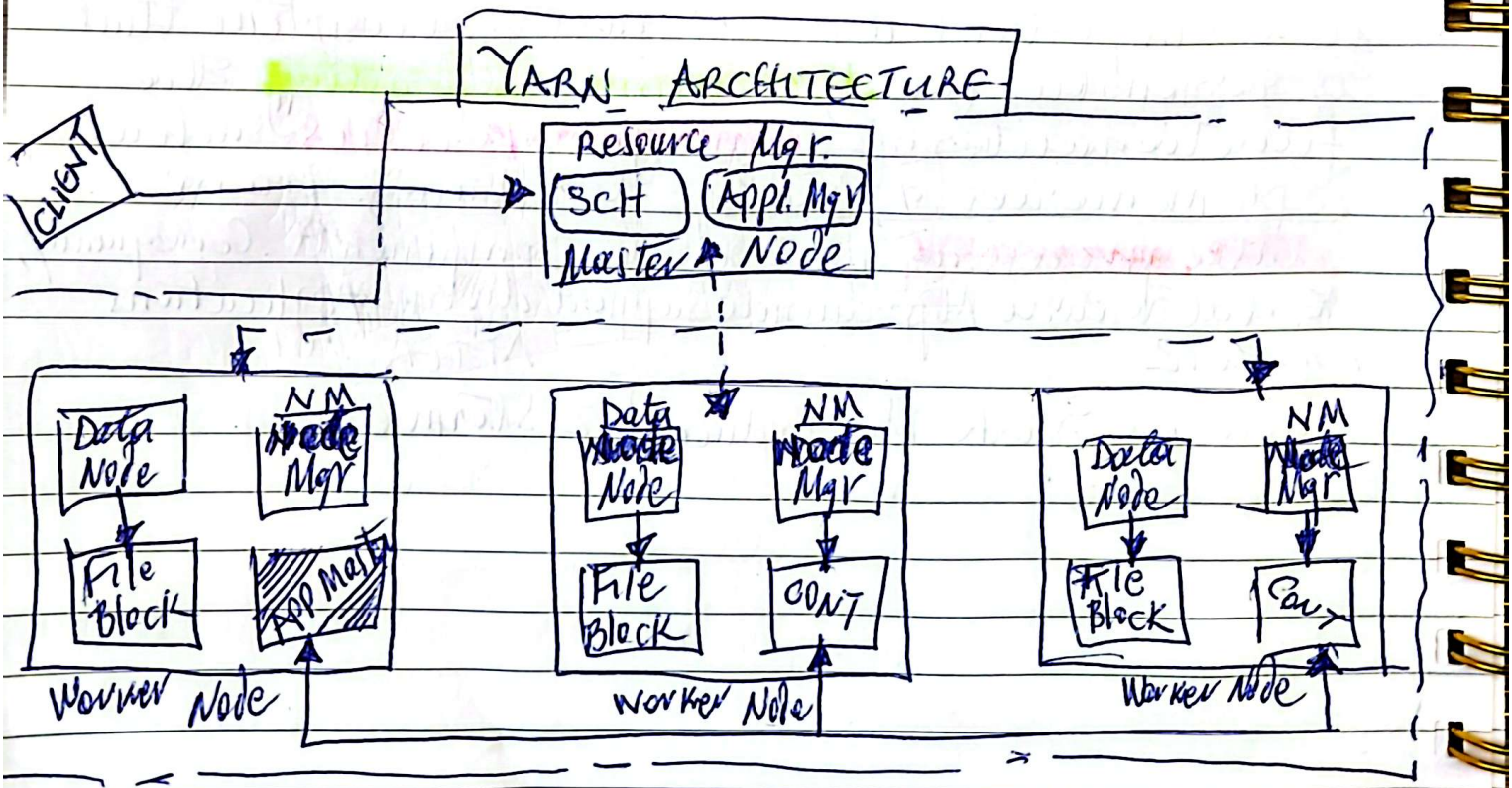
Map Reduce (D.P) | Spark, Storm, Tez (D.P)

YARN
(Cluster Resource Mgt)

HDFS
Redundant/Reliable Storage

Benefit.

1) Centralized Resource Management
2) Flexibility of Application across the cluster/ Framework.
3) Cost Effective: All in one Framework.
4) Better data utilization Amongst Applications.
5) Less Data Movement within same framework/cluster Applications.



Components of YARN
Client, Resource Mgr, Scheduler, Application Mgr, Data Node, File Block, Node Mgr, Container, Application Master, (~~Master Worker~~ & ~~Node Worker~~.)
- Master Node & Worker Node

HIVE

Apache Hive is a Data Warehousing Infrastructure built on top of Hadoop to provide:
* SQL-like Query language called Hive Query lang. (HQL)
* Structured data analysis on large Datasets stored in HDFS.
* Batch processing rather than Real-time transactions. This batch processing is also referred to as Write-Once-Read-Many (WORM).

~~Why~~ Use Hive.
- ~~Familiarity~~ to SQL-like interface (HQL) makes it easy
- Scalability & fault tolerance by running Queries using MapReduce or Tez.
- Supports multiple formats e.g text, ORC, Parquet, Avro
- Efficient query ~~partition~~ Execution using partition & Indexing.

Main components of Hive.
1) Hive Client: Interfaces like CLI, JDBC, ODBC drivers & Beeline
2) Hive Services:
   - Drivers: parses Queries, execution plans
   - Metastore: Stores Metadata (Schema, Partition, locations)
   - Execution Engine: Converts Queries into MapReduce,
3 Hive Storage (HDFS) Tez or Spark jobs.
   Stores actual Data files.

Execution flow of a Query.

1. Query Submission via CLI, Beeline or JDBC
2. Query parsing and compilation
3. Execution Plan generation (MapReduce/Tez)
4. Query Execution & result retrival.

Primitive Data Types.

Int, Bigint, Float, Double, Boolean, String & Date/TimeStamp

Complex Data type.

Array, Map, Struct.
Array = Array(int)
Map = Map(String, int)
Struct = Struct(name:String, age:Int)

Syntaxes / Statements Common.

(i) Create database mydb;
(ii) Use mydb;
(iii) Create table employees (
id int,
name String,
salary float
) Row format delimeted
field terminated by ',';

For EXTERNAL TABLE
Create External Table
location: ' ';

(iv) Insert into employees
Values (1, 'John', 5000)
(2, 'Kelvin', 6000)
(3, 'curry', 7000);

(v) Select * from employees;

Working
Managed Ta
Create tabl
(id INT, nam

loading
hdfs dfs
LOAD DAT
INTO TAB

Partitio
Partition
Create
i
p

Bucketi
Create
id
) cl

Hive F
String
Date fu
Aggregate

Perfor
+ Use
- 10 e
Exe
⇒ S

Enab
⇒ Se

Working with Tables:

Managed Table by Hive	External table by 3rd party
Create table tablename (id INT, name String);	Create External table tablename (id int, name String) LOCATION 'hdfs dfs/path/';

loading Data

hdfs dfs - put employees.text /user/hive/warehouse/
LOAD DATA PATH '/user/hive/warehouse/employees.text'
INTO TABLE employees;

Partitioning & Bucketing

Partitioning (Improves Query Performance)
Create table sales (
id int, amount float)
partitioned by (year int);

Bucketing (Distributes data into files).
Create table users (
id int, names String
) clustered by (id) into 4 buckets;

Hive Functions.

String function: Select Upper (name), Lower (name from Emp;
Date function: Select current_date, Year ('2025-03-18') from Emp;
Aggregate fn: Select count(*), Sum (Salary) from Emp;

Performance Optimization

- Use ORC, or Parquet formats instead of txt.
- To enable partitioning & bucketing, use Vector Query Execution.
  ⇒ Set hive.vectorized.Execution.enabled = true;

Enable Cost Based Optimization (CBO)
⇒ set hive.cbo.enabled = true;

Hive Integration with Other tools include:
- Apache Spark (SparkSQL can run on Hive)
- Apache HBase (Hive can Query NoSQL data)
- Apache Airflow (for Job Scheduling).

Hive Applications with Real-World Use Cases:
- Log Analysis (e.g Web Server logs)
- ETL pipelines (Extract-Transform-load process)
- Data Warehousing (Aggregations, Reporting).