1. Hive

=> Hive is a data warehouse infrastructure tool to process structured data in Hadoop

=> it resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy

=> Initially, Hive was developed by facebook, Later the Apache software foundation took it up

=> it is used by different companies

=> Amazon uses it in Amazon Elastic MapReduce

Hive is not

=> A relational database

=> A design for online Transaction Processing (OLTP)

=> A language for real-time queries and row-level updates

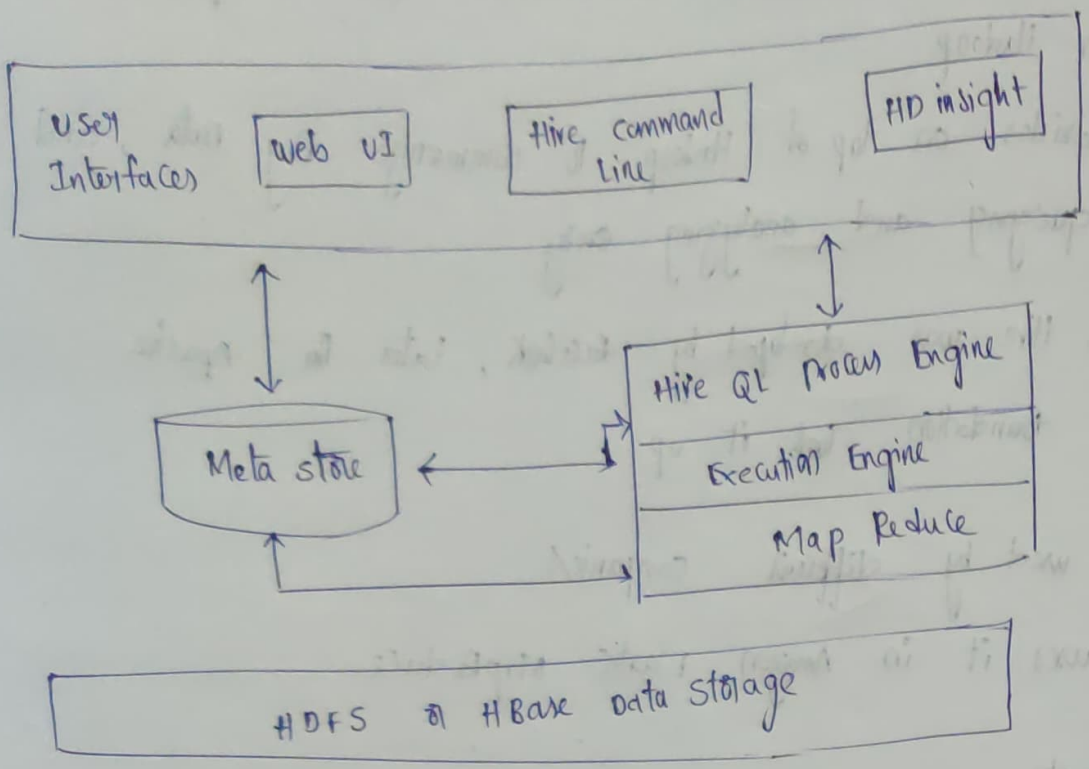
Features

=> it stores schema in a database and processed data into HDFS.

=> it is designed for OLAP

=> it provides SQL type language for querying called HiveQL or HQL

Architecture of Hive



User Interface

* Hive is a data warehouse infrastructure software that can create interaction b/t user and HDFS
* The user interfaces that Hive supports are Hive web UI, Hive Command Line and Hive HD insight (In windows Server).

Meta store

* Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table

-> But Not tables.
-> Meta store stores in RDBMS

HiveQL Process Engine

- HiveQL is similar to SQL
- it is one of the replacements of traditional approach for MapReduce program
- Instead of writing MapReduce program in java, we can write a query for MapReduce job and process it

Execution Engine

- The ~~Conclusion~~ Conjunction part of HiveQL process Engine and Map Reduce is Hive Execution Engine
- Execution Engine processes the query and generates results as same as MapReduce results
- It uses the flavor of MapReduce.

HDFS or HBase

- Hadoop distributed file system or HBase are the data storage techniques to store data into file system

Different modes of Hive

1. Local mode - (When to use)

* if the Hadoop installed under pseudo mode with having one data node we use Hive in this mode
* if the data size is smaller in term of limited to signal local machine.

2. Map reduce Mode (When to use).

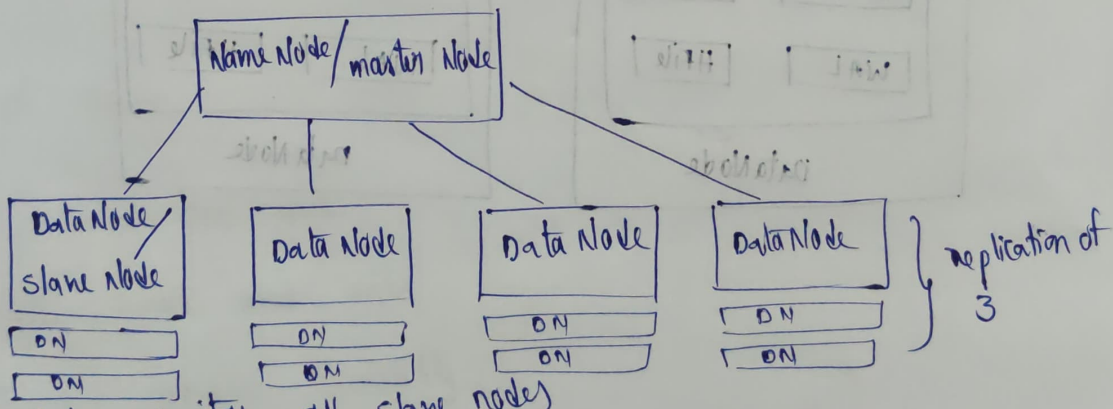
* if Hadoop is having multiple data nodes and data is distributed across different node we use Hive in this mode

HBase

- HBase is built over HDFS file system
- NOSQL Data store
- HBase can use MapReduce to retrieve or store Data.
- HBase is column family oriented.

what is HDFS?

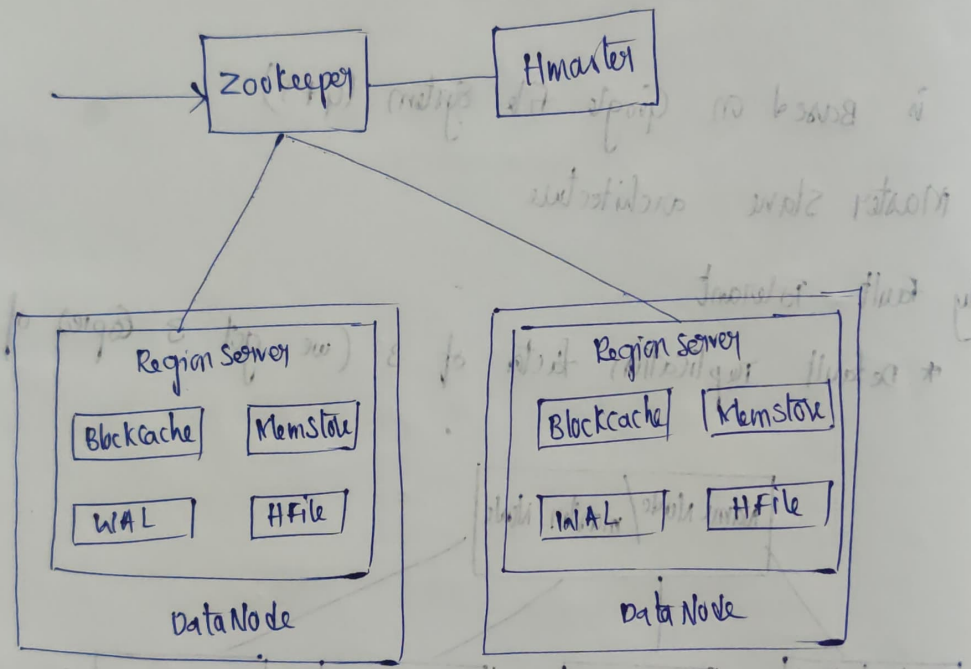
- HDFS stands for Hadoop Distributed file system
- HDFS is used to scale single cluster to 100's or 1000's of nodes
- HDFS is Based on Google file system (GFS)
- uses Master slave architecture
- Highly fault Tolerant
  - * Default replication factor of 3 (we get 3 copies of Data)



- Master Node Monitors all slave nodes
- * Slave nodes stores the Data
- Only one Master Node
- * Multiple Slave Nodes

HBase Architecture

* HBase we have Hmaster, and we can use multiple Hmaster but only one HMaster is active at a time
* if current Hmaster will failed due to some issue, then next Hmaster will be active.
* Here, we have Region servers like Data nodes in HDFS
* Zookeeper tracks the all region servers and H master



Storage Mechanism in HBase

- HBase is a column-oriented database and the tables in it are sorted by row
- The table schema defines only column families, which are the key value pairs.
- A table have multiple column families and each column family can have any number of columns.
- Each cell value of the table has a timestamp.

HBase Table Components

1. Table
2. Row
3. column family
4. column
5. Timestamp

Features of HBase

- Atomic read and write
- Consistent reads and writes
- Linear and modular scalability
- Automatic and configurable sharding of tables
- Easy to use java API for client access.
- it provides data replication across clusters

Zookeeper ?

- Zookeeper is basically keeps track of information that must be synchonized across your cluster
  - which node is the master ?
  - what tasks are assigned to which workers ?
  - which workers are currently available ?

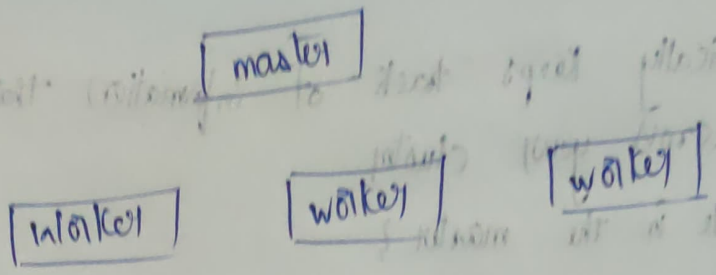
- Zookeeper is a tool that applications can use to recovery from partial failures in your cluster.
- Zookeeper is an integral part of - HBase.
  - HA (High Availability) Map reduce
  - Drill
  - storm .. etc

failure modes

1. master crashes
   - if any master will crashes then zookeeper will track the master node and place/active the new master node.
   - Only one masternode will active at a time

2. worker crashes
   - our data is stored in worker nodes,
   - if worker node will down, zookeeper have another copy of the same workernode.

3. Network trouble
   - part of your cluster can't see the rest of it



Primitive operations

1. master election

+ One node registers itself as a master, and holds a "lock" on that data
+ other nodes cannot become master until that lock is released
+ only one node allowed to hold the lock at a time

2. Crash detection

+ "Ephemeral" data on a node's availability automatically goes away if the node disconnects, or fails to refresh itself after some time out period.

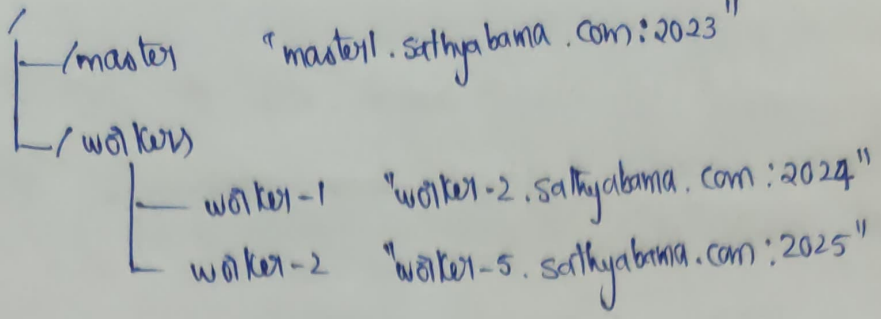
3. Group management

4. Metadata — list of outstanding tasks, task assignments

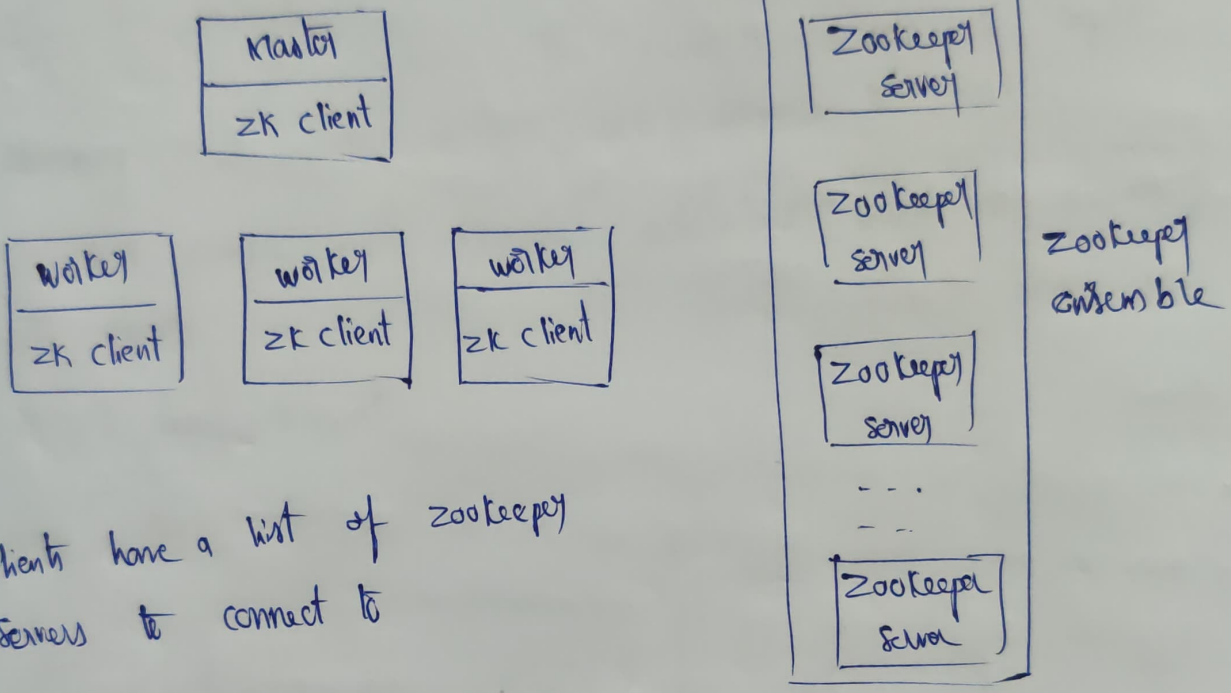
Zookeeper's API

- create
- delete
- exists
- setData
- getData
- getChildren

example



Zookeeper Architecture



* Clients have a list of zookeeper servers to connect to

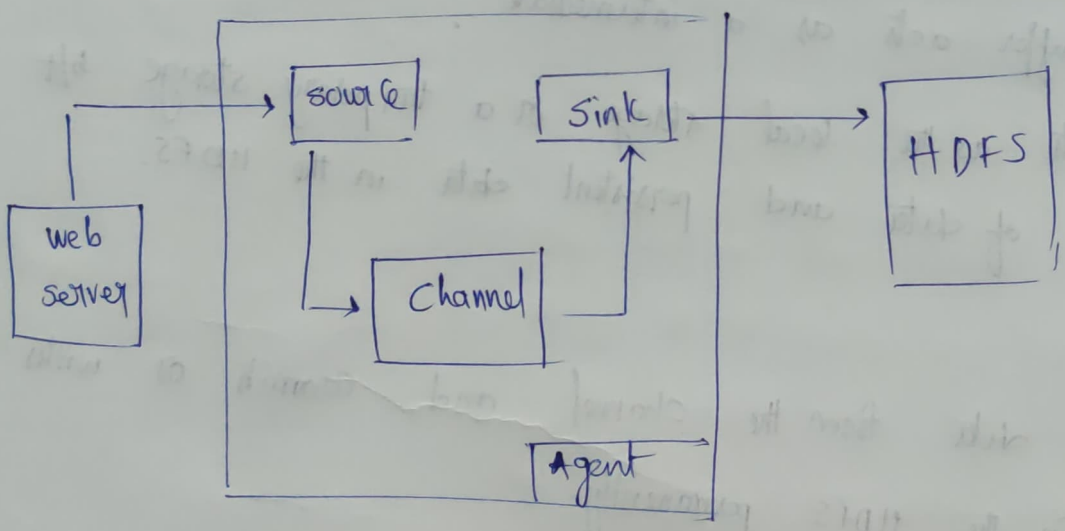
Flume

- Flume is a tool for data ingestion in HDFS
- it collects, aggregates and transports large amount of streaming data such as log files, events from various sources like Network traffic, social media, email messages etc. to HDFS
- flume is a highly reliable & distributed

Basic idea behind Flume

- The main idea behind the flume's design is to capture streaming data from various web servers to HDFS.
- it has simple and flexible architecture based on streaming data flows.
- it is fault-tolerant.

Flume Architecture



- flume agent which ingests the streaming data from various data sources to HDFS.

* From the diagram, we can easily understand that the web server indicates the data source.
* Twitter is among one of the famous sources for streaming data.
* The flume agent has 3 components
  - source
  - sink
  - channel

source:

* it accepts the data from the incoming streamline and stores the data in the channel.

channel:

* In general, the reading speed is faster than the writing speed.
* Thus, we need some buffer to match the read & write speed difference. buffer acts as a intermediate
* channel acts as the local storage or a temporary storage b/t the source of data and persistent data in the HDFS.

Sink:

* Collects the data from the channel and commits or writes the data in the HDFS permanently.

Advantages

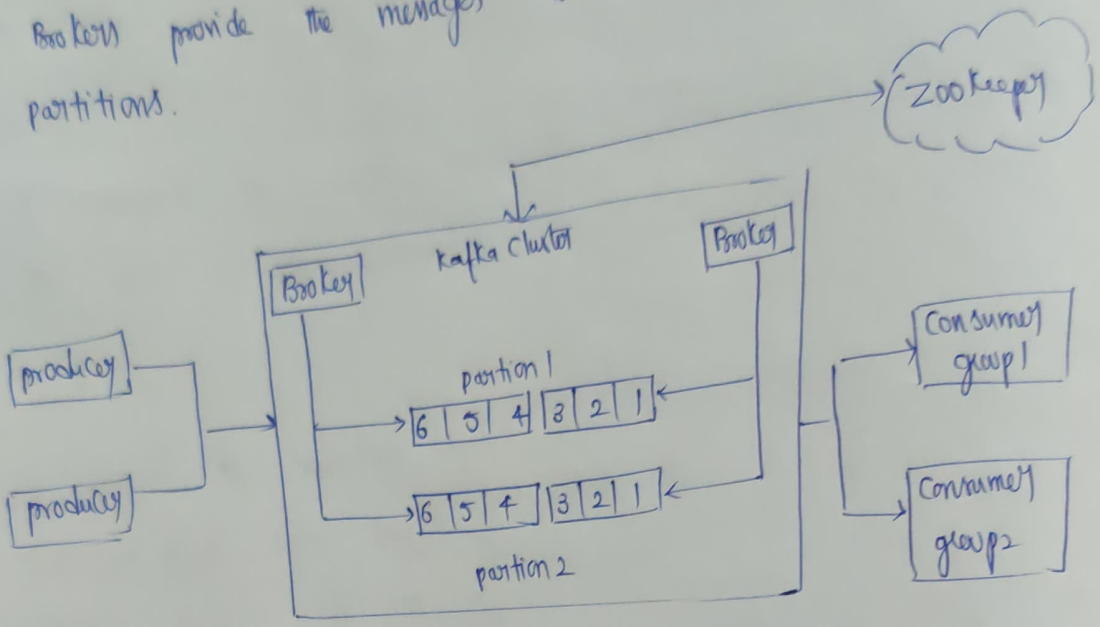
- Flume is scalable, reliable, fault tolerant.
- Flume is customizable for different sources and sinks
- Flume can store data in centralized stores like HBase & HDFS
- Flume is horizontally scalable.
- Using Flume, we can ingest data from multiple servers into Hadoop.

Kafka

- Kafka is a high-performance, real-time messanging system
- it is an open source tool and is a part of Apache projects

Architecture

- Kafka architecture consists of brokers that take messages from the producers and add to partition of a topic.
- Brokers provide the messages to the consumers from the partitions.

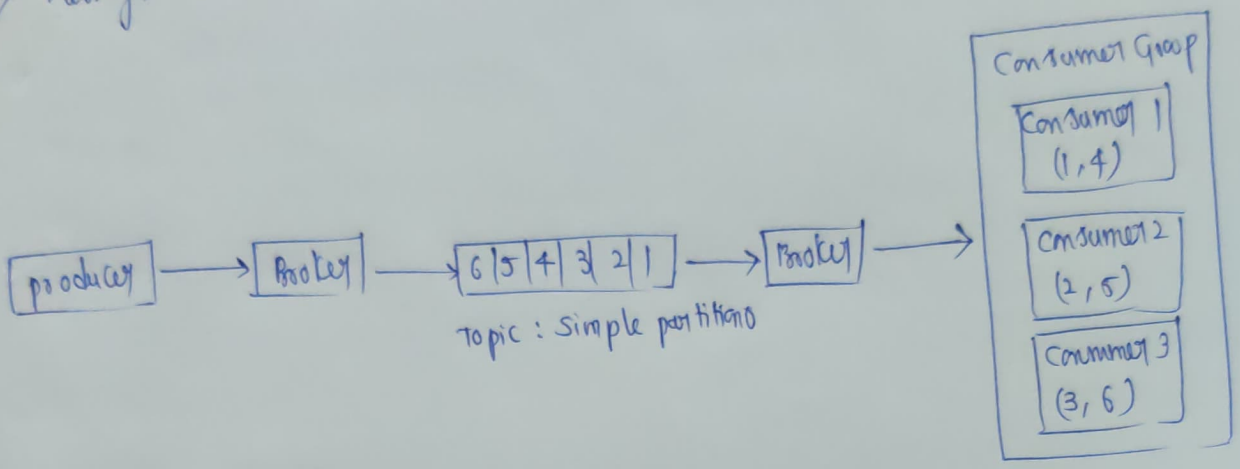


- A topic is divided into multiple partitions
- The messages are added to the partitions at one end and consumed in the same order.
- Each partition acts as a message queue.
- Consumers are divided into Consumer groups
- Each message is delivered to one Consumer in each Consumer group
- Zookeeper is used for coordination.

Kafka architecture supports the publish - subscribe and queue system

1. Queue System

- Each message has to be consumed by only one Consumer
- Each message is consumed by any one of the available consumers
- Messages are consumed in the same order that they are received



2. publish - subscribe

- Each message is received by all the subscribers
- Each Subscriber receives all the messages
- Messages are received in the same order that they are produced.