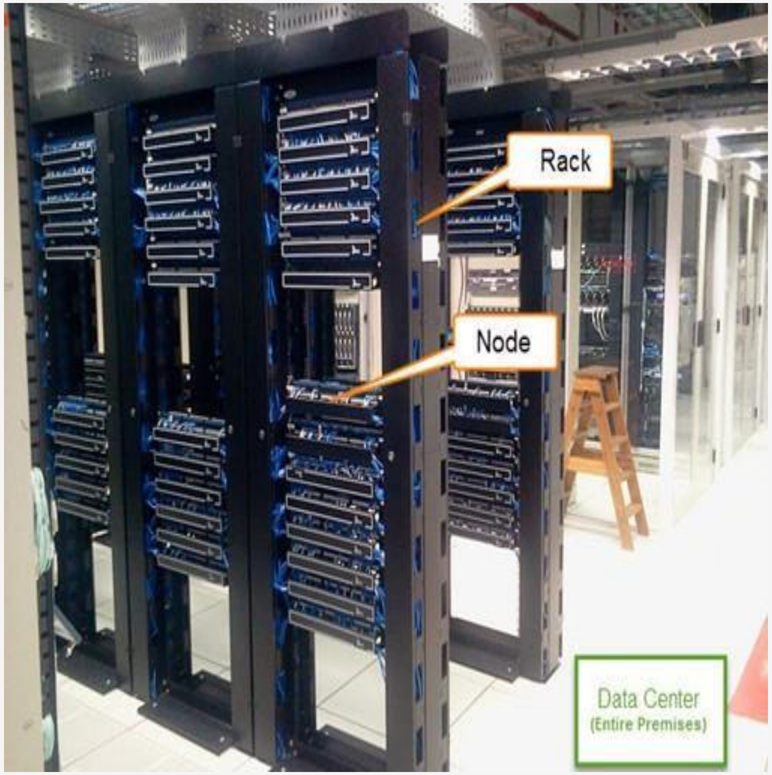
Architecting Hadoop

Chapter - 03

Network Topology In Hadoop

- **Nodes**
- **Racks**
- **Cluster**



Master Slave Architecture of Hadoop

- Hadoop has a Master-Slave Architecture for data storage and distributed data processing using MapReduce and HDFS methods.
- **Master node:**
  - **NameNode, Job Tracker**
  - The master node allows you to conduct parallel processing of data and manages the storage components
  - Generally one Per Cluster
- **Slave node:**
  - **DataNode, Task Tracker**
  - The slave nodes are the additional machines in the Hadoop cluster which allows you to store data to conduct complex calculations. Moreover, all the slave node comes with Task Tracker and a DataNode. This allows you to synchronize the processes with the NameNode and Job Tracker respectively.
  - As many as you decide as per the Scalability.

In Hadoop, master or slave system can be set up in the cloud or on-premise

Hadoop Daemons for Rel. 1 Operations

Namenode

- The **namenode** daemon is a master daemon and is responsible for storing all the location information of the files present in HDFS. The actual data is never stored on a namenode. In other words, it holds the metadata of the files in HDFS.

Secondary namenode

- The **secondary namenode** daemon is responsible for performing periodic housekeeping functions for namenode.

Data Node

- The **datanode** daemon acts as a slave node and is responsible for storing the actual files in HDFS.

Hadoop Daemons for Rel. 1 Operations

Jobtracker

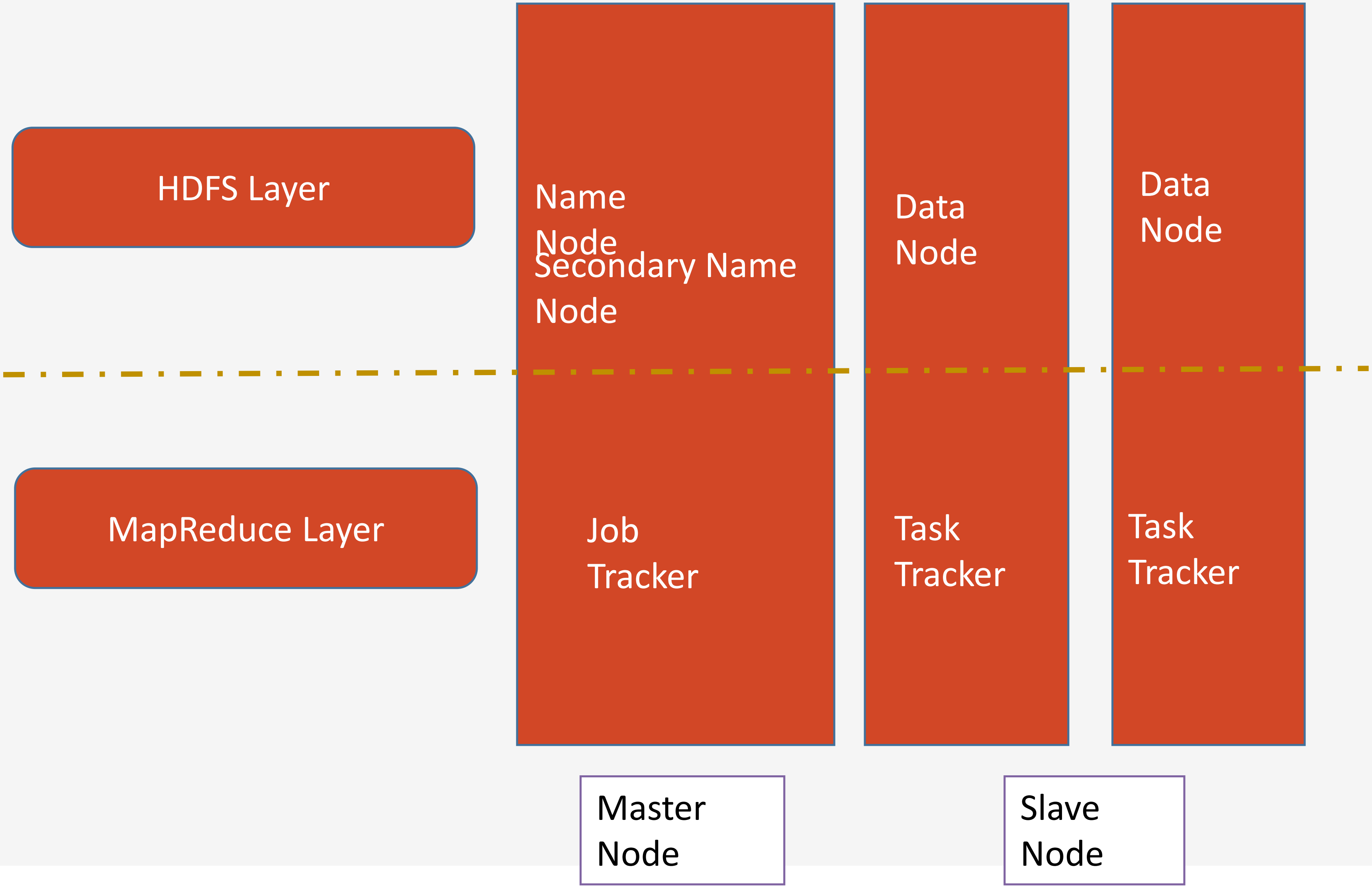
- The **jobtracker** daemon is responsible for accepting job requests from a client and scheduling/assigning tasktrackers with tasks to be performed. The jobtracker daemon tries to assign tasks to the tasktracker daemon on the datanode daemon where the data to be processed is stored.

Tasktracker

- The **tasktracker** daemon is a daemon that accepts tasks (map, reduce, and shuffle) from the jobtracker daemon. The tasktracker daemon is the daemon that performs the actual tasks during a MapReduce operation. The tasktracker daemon sends a heartbeat message to jobtracker, periodically, to notify the jobtracker daemon that it is alive. Along with the heartbeat, it also sends the free slots available within it, to process tasks. The tasktracker daemon starts and monitors the map,

NOTE :
In small clusters, the namenode and jobtracker daemons reside on the same node.
However, in larger clusters, there are dedicated nodes for the namenode and jobtracker daemons.

Master Slave Architecture of Hadoop – Rel. 1



HDFS Layer

MapReduce Layer

Name Node

Secondary Name Node

Job Tracker

Data Node

Task Tracker

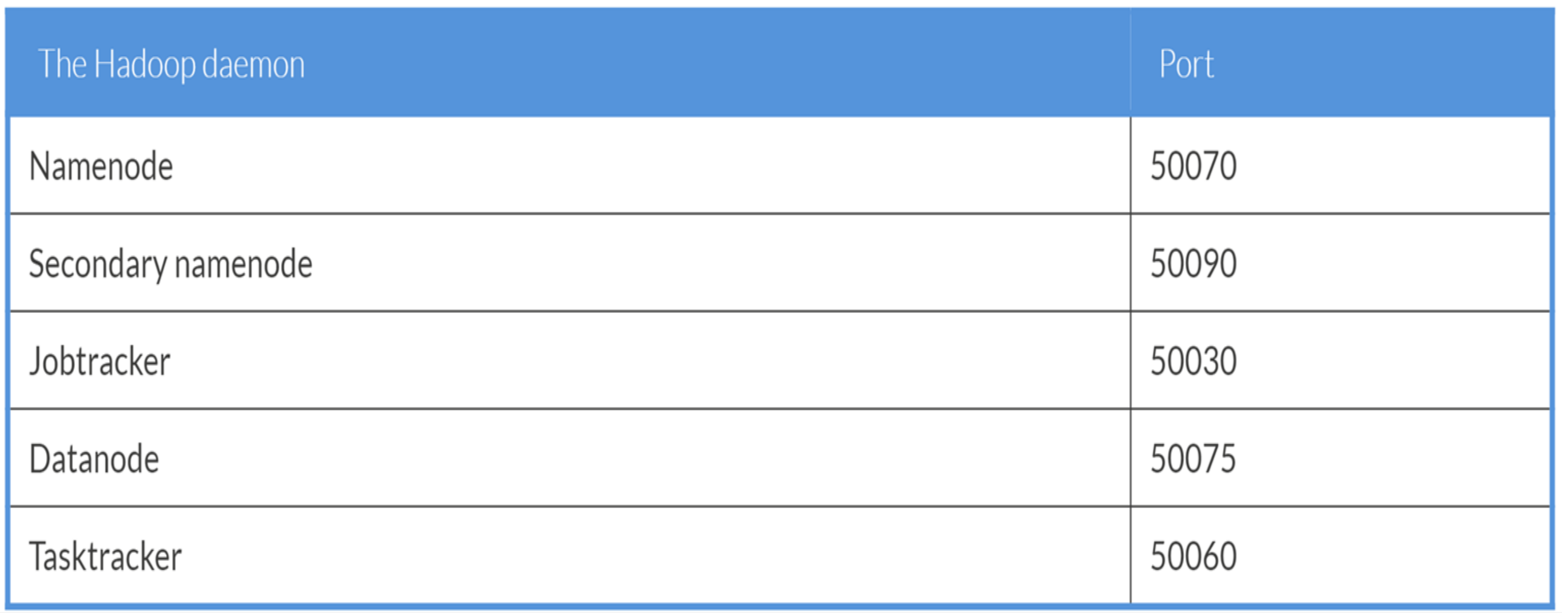
Data Node

Task Tracker

Master Node

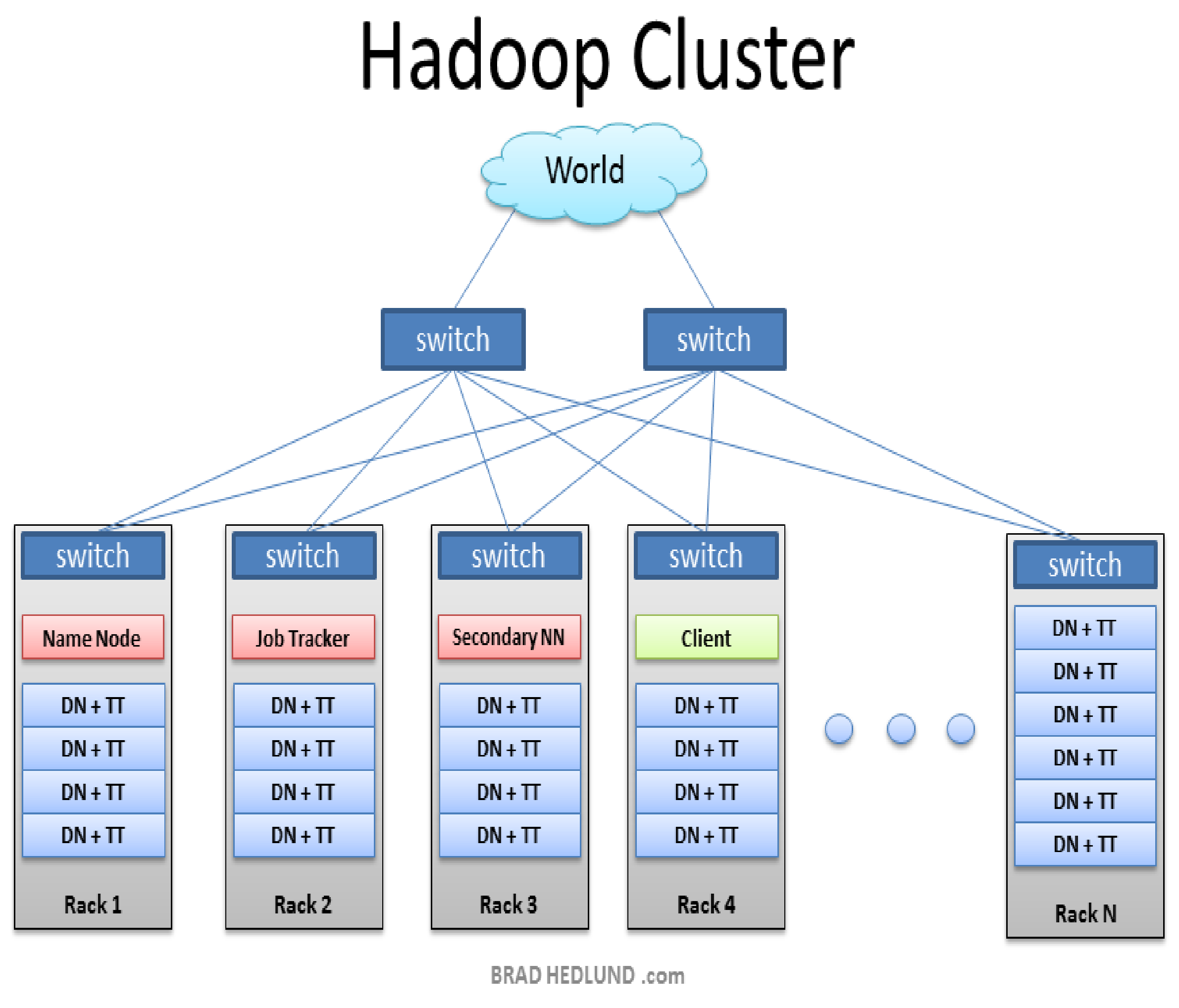
Slave Node

OLAP vs OLTP



The Hadoop daemon	Port
Namenode	50070
Secondary namenode	50090
Jobtracker	50030
Datanode	50075
Tasktracker	50060

Network Topology In Hadoop



OLAP vs OLTP

OLTP (Online Transaction Processing) -

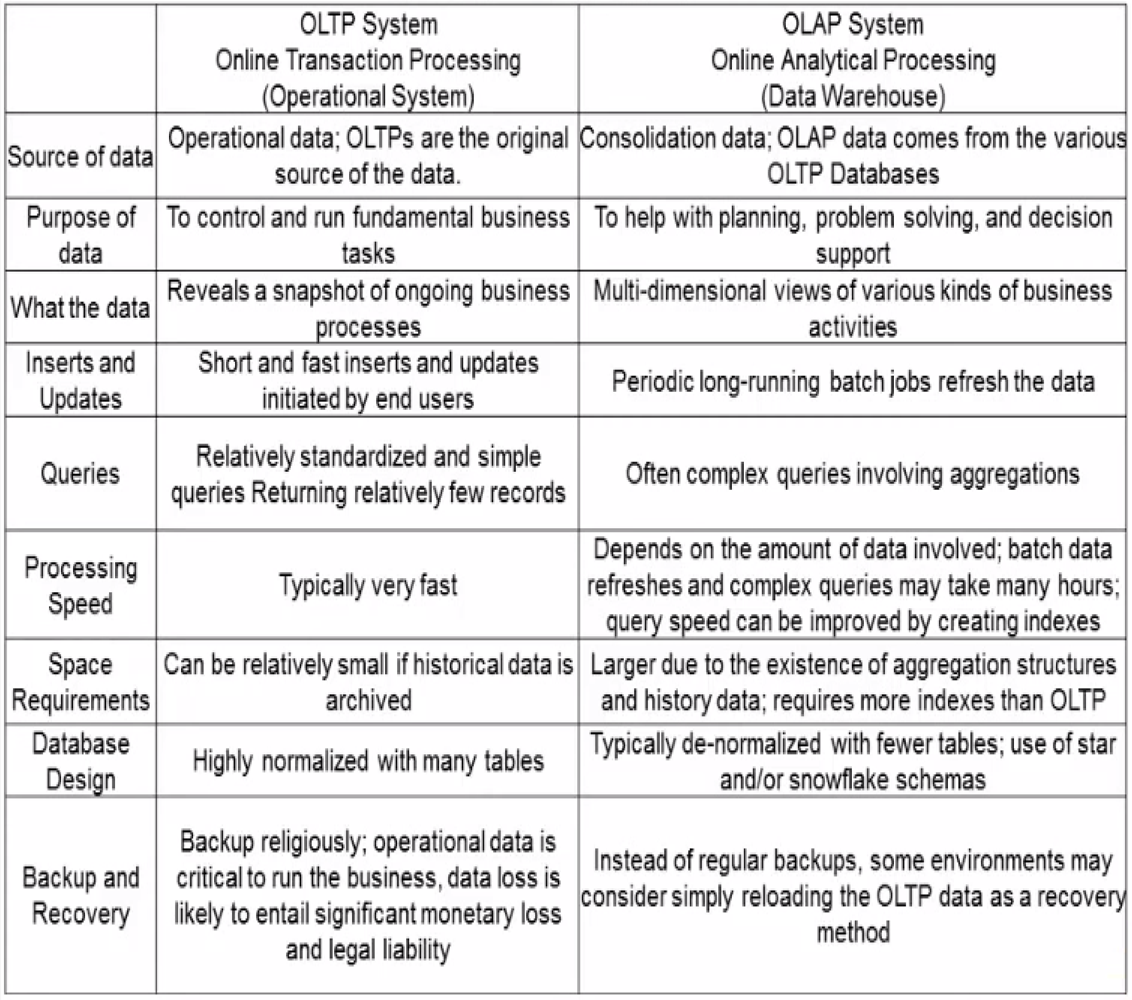
Processes a large number of short online transactions (INSERT, UPDATE, DELETE). Fast query processing while maintaining data integrity in multi-access environments. The effectiveness is measured by number of transactions per second.

TBD

OLAP (Online Analytical Processing) -

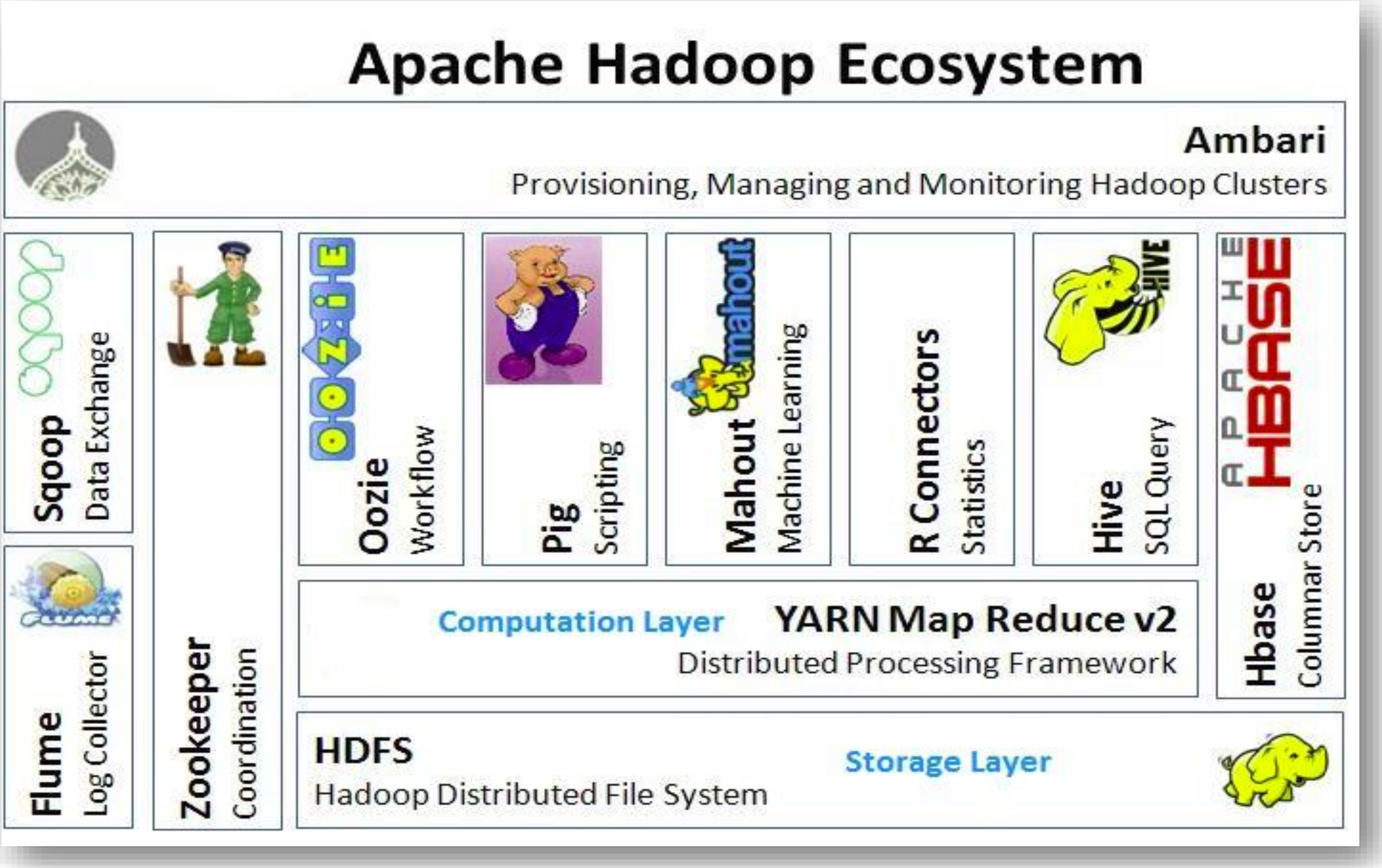
Processes relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas such as star schema.

Hadoop is ideal for OLAP kind of data systems.



	OLTP System Online Transaction Processing (Operational System)	OLAP System Online Analytical Processing (Data Warehouse)
Source of data	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries Returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas
Backup and Recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

Hadoop Ecosystem Tools



Apache Hadoop Ecosystem

Ambari
Provisioning, Managing and Monitoring Hadoop Clusters

Sqoop
Data Exchange

Flume
Log Collector

Zookeeper
Coordination

Oozie
Workflow

Pig
Scripting

Mahout
Machine Learning

R Connectors
Statistics

Hive
SQL Query

Hbase
Columnar Store

Computation Layer

YARN Map Reduce v2
Distributed Processing Framework

HDFS
Hadoop Distributed File System

Storage Layer

Prerequisites to Learn Hadoop

Hadoop Based on 2 Technologies:

- Core Java
- SQL

- Core Java can be learnt / brushed up from tutorials:
- The new Boston (Tutorial 1 to Tutorial 61) : 8 Hours Video Lectures
- https://thenewboston.com/videos.php?cat=31
- https://www.youtube.com/playlist?list=PLFE2CE09D83EE3E28