Hadoop

A software framework employed for clustered file system and handling of big data.

Runs on commodity hardware in an existing data center.

It can run on a cloud infrastructure.

Hadoop consists of four parts :

- Hadoop Distributed File System: Commonly known as HDFS, it is a distributed file system compatible with very high scale bandwidth.
- MapReduce: A programming model for processing big data.
- YARN: It is a platform used for managing and scheduling Hadoop's resources in Hadoop infrastructure.
- Libraries: To help other modules to work with Hadoop.

Hadoop

Pros :

- The core strength of Hadoop is its HDFS which has the ability to hold all type of data video, images, JSON, XML, and plain text over the same file system.
- Highly useful for R&D purposes.
- Provides quick access to data.
- Highly scalable
- Highly-available service resting on a cluster of computers

Cons :

- Sometimes disk space issues can be faced due to its 3x data redundancy.
- I/O operations could have been optimized for better performance.

HDFS

- Hadoop Distributed File System is the core component or the backbone of Hadoop Ecosystem.
- HDFS is the one, which makes it possible to store different types of large data sets. HDFS creates a level of abstraction over the resources, from where one can see the whole HDFS as a single unit.
- It helps in storing data across various nodes and maintaining the log file about the stored data (metadata).
- HDFS has two core components, i.e. NameNode and DataNode.
  1. The NameNode is the main node and it doesn’t store the actual data. It contains metadata, just like a log file. Therefore, it requires less storage and high computational resources.
  2. On the other hand, all the data is stored on the DataNodes and hence it requires more storage resources. These DataNodes are commodity hardware in the distributed environment.
  3. User communicate to the NameNode while writing the data. Then, it internally sends a request to the client to store and replicate data on various DataNodes.

Map Reduce

It is the core component of processing in a Hadoop Ecosystem as it provides the logic of processing.

In other words, MapReduce is a software framework which helps in writing applications that processes large data sets using distributed and parallel algorithms inside Hadoop environment.

In a MapReduce program, Map() and Reduce() are two functions.

1. The Map function performs actions like filtering, grouping and sorting.
2. While Reduce function aggregates and summarizes the result produced by map function.
3. The result generated by the Map function is a key value pair (K, V) which acts as the input for Reduce function.

YARN

Yet Another Resource Negotiator

– Resource allocation and scheduling

– Core Hadoop component

■ Components: ResourceManager, NodeManager

– ResourceManager:

■ receives processing requests

■ passes the parts of requests to corresponding NodeManagers

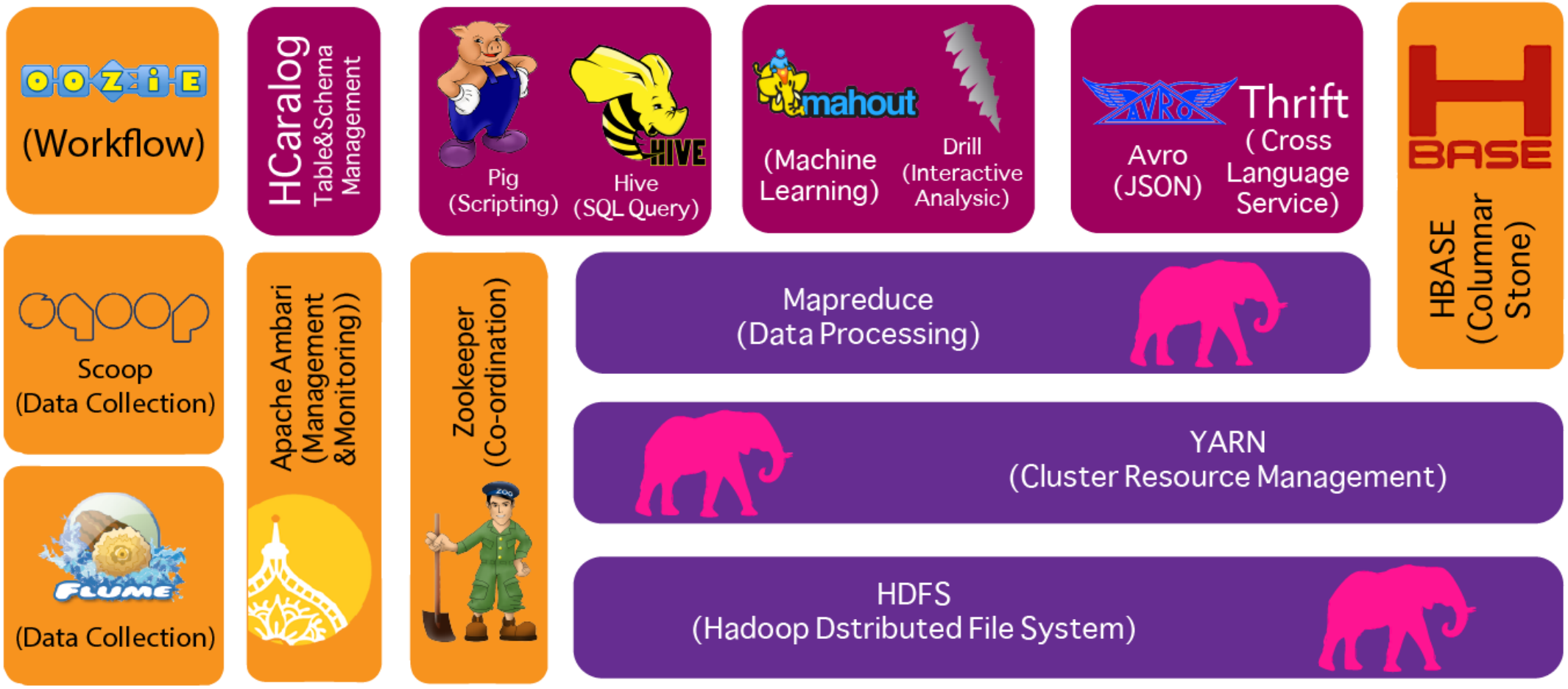
■ Has Schedulers that allocate resources, time based on application requirements

■ Has ApplicationsManager that monitors running jobs

– NodeManager:

■ Handles requests at every DataNode

HADOOP ECOSYSTEM



Apache Pig

- **SQL-like** command structure in Hadoop
  - Much more condensed (10 pig latin lines ≈ 200 Map-Reduce lines)
  - Allows actions like grouping, filtering etc.
  - Developed by Yahoo

- **Pig Runtime** and **Pig Latin** language
  - Analogy to Java: Pig Runtime -> JVM, Pig Latin -> Java
  - **Compiler** internally converts pig latin to MapReduce

Apache HIVE

- **SQL** queries in Hadoop:
  - Uses Hive Query Language(HQL), very similar to SQL
  - Highly scalable, both batch and real-time processing support
  - Supports all SQL types, most commands etc.

- **JDBC/ODBC driver** and **Hive Command Line** :
  - **Java Database Connectivity (JDBC), Object Database Connectivity (ODBC)**
    - Used to establish connection with data storage
  - Developed by Facebook

Apache Mahout

- **Machine Learning** in **Hadoop**
  - Provides built-in algorithms for machine learning problems
  - Executed through a command line

- Supported algorithms:
  - **Collaborative filtering**: mining patterns/behaviors, makes predictions and recommendations
    - Amazon product recommendation
  - **Clustering**: finding groups of similar data
    - recommending groups in social media
  - **Classification**: classifying and categorizing data into various sub-departments
    - identifying objects in image recognition

Apache HBASE

- **Non-relational** distributed database (**No-SQL**)
  - All types of data, absolutely everything is supported
  - Provides fault tolerance and fast retrieval of data
  - Open source, based on Google's BigTable

- Runs on top of Hadoop, provides BigTable - like capabilities
  - Written in Java

Apache Zookeeper, Oozie

- **Zookeeper**: Hadoop **job coordination**
  - Coordination between different **distributed** Hadoop **jobs/services**
  - Things like addresses, start-up/shutdown, configurations
  - Used in Rackspace, Yahoo, eBay

- **Oozie**: Hadoop clock/alarm
  - **Oozie Workflow**: sequential acts to be performed
  - **Oozie Coordinator**: triggers job execution when data is available

Apache Flume, Sqoop

- **Flume: Unstructured** data **ingestion**
  - Handles the entry of data in the system
  - **Collects, aggregates** and **moves** large amounts of data
  - Handles **real-time input streams**

- **Sqoop**: **Import/export structured** data
  - Also handles data ingestion
  - Moves data from **RDBMS** or **Enterprise** data warehouses to **HDFS** or vice versa

Apache Solr & Lucene

- **Searching** and **indexing**
  - Used for different data search tasks
  - Solr is the application, Lucene is the engine/kernel

Apache Ambari

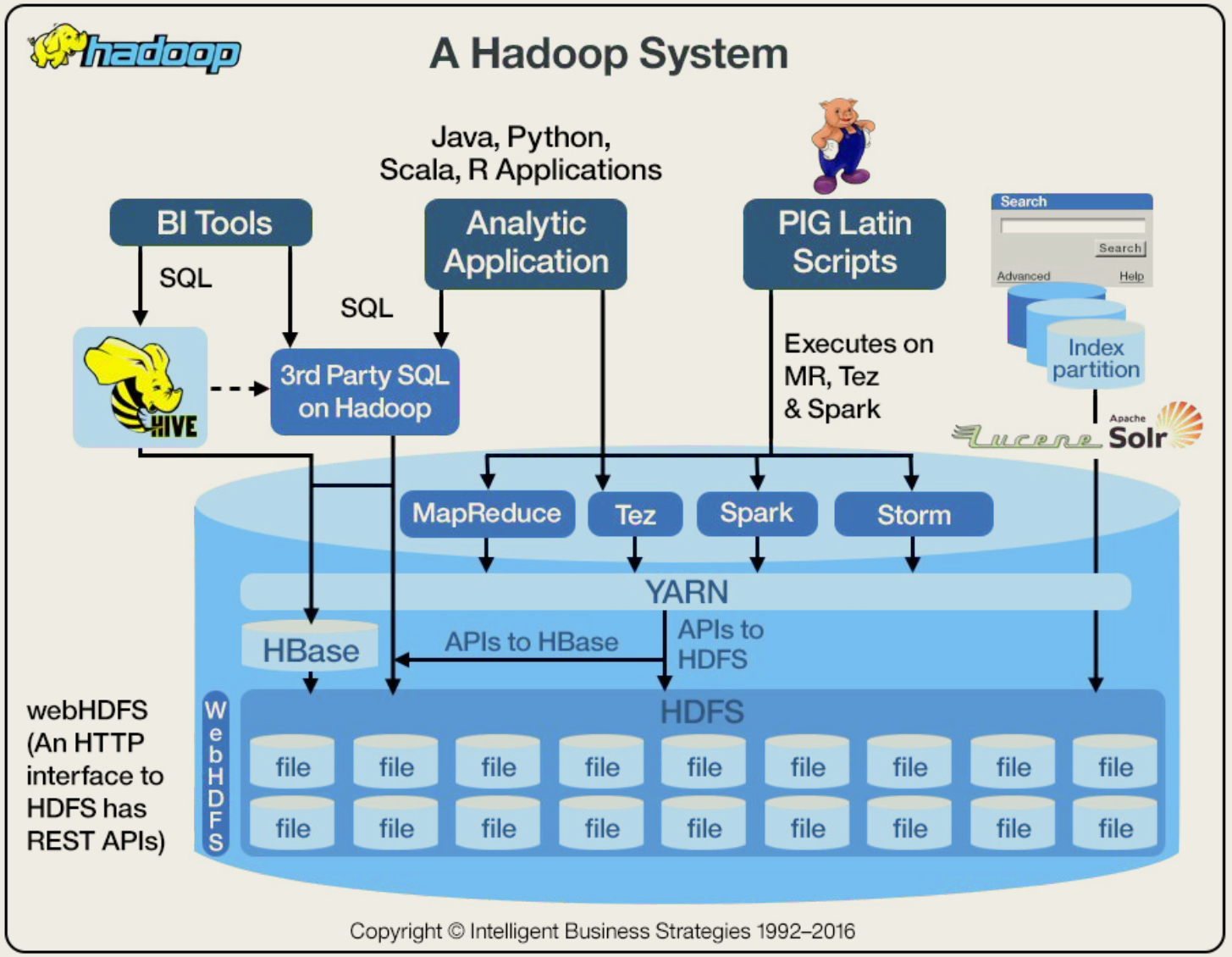
- **Managing** the **whole ecosystem**

- **Hadoop cluster provisioning**
  - Step by step process for installing hadoop on many hosts
  - Handles Hadoop cluster configurations
- **Hadoop cluster management**
  - Provides central management service for starting, stopping and re-configuring Hadoop services
- **Hadoop cluster monitoring**
  - Dashboard for monitoring cluster health and status
  - Amber Alert framework for notifying if something is wrong

Honorable mentions

- **Avro**: data serialization (~JSON)
- **Cassandra**: reliable NoSQL distributed database
- **Cloudera**: Hadoop environment management, commercial vendor
- **Chukwa**: data collection system
- **Impala**: analytic database
- **Kafka**: Hadoop messaging
- **Tajo**: robust big data relational and distributed data warehouse
- **Tez**: generalized data-flow programming framework

An example Hadoop system



Thank you!

Based on:

https://www.edureka.co/blog/hadoop-ecosystem
http://www.bmc.com/guides/hadoop-ecosystem.html