Big Data Analytics(Hadoop)

Prepared By : Manoj Kumar Joshi & Vikas Sawhney

General Agenda

- Understanding Big Data and Big Data Analytics
- Getting familiar with Hadoop Technology
- Hadoop release and upgrades
- Setting up a single node hadoop cluster

Acknowledgement

- Thanks to all the authors who left their self-explanatory images on the internet.
- We own the errors of course

What is Big Data?

- Big data is a buzzword, or catch-phrase, used to describe a massive volume of both structured and unstructured data that is so large that it's difficult to process using traditional database and software techniques.
- Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone.
- Big Data defined as high volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.
- According to IBM, 80% of data captured today is unstructured, from sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals, to name a few. All of this unstructured data is Big Data.

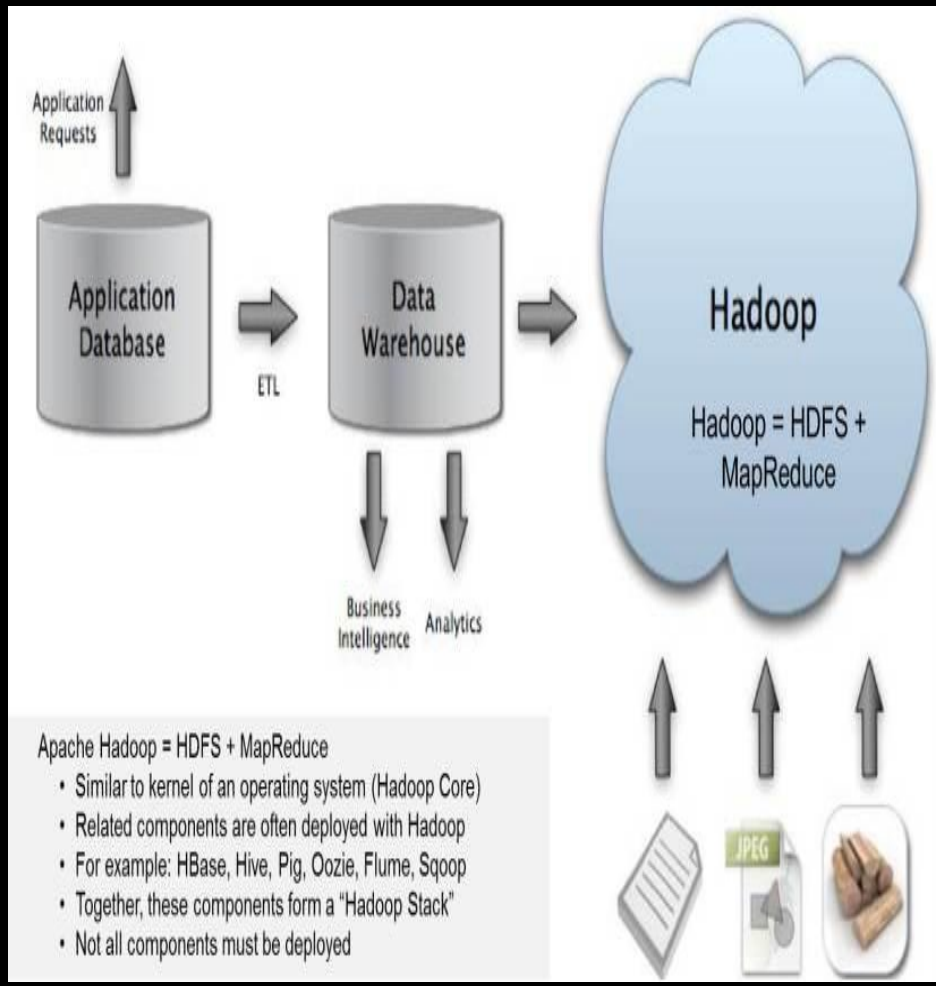
Benefits of Big Data Analytics

The real issue is not that you are acquiring large amounts of data. It's what you do with the data that counts. The hopeful vision is that organizations will be able to take data from any source, harness relevant data and analyze it to find answers that enable :

- Cost reductions,
- Time reductions,
- New product development and optimized offerings
- Smarter business decision making

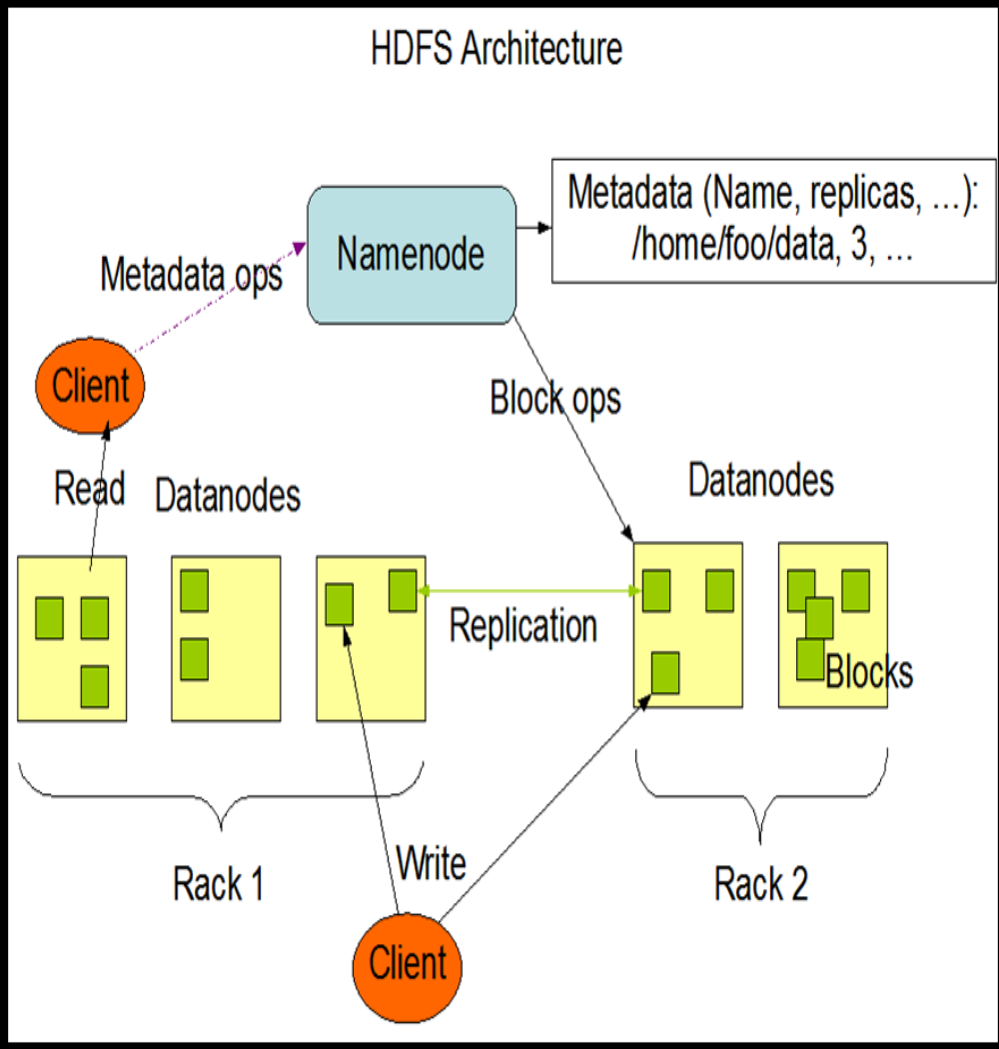
What is Hadoop (Recap)

- Open source software platform for scalable, distributed computing
- Two primary components :
  - HDFS : Distributed file system provides economical, reliable, fault tolerant and scalable storage of very large datasets across machines in the cluster
  - MapReduce : A programming model for efficient distributed processing. Designed to reliably perform computation on large volume of data in parallel



Hadoop Architecture

- **Name Node** : Keeps the metadata of all files/blocks in the file system, and tracks where across the cluster the file data is kept
- **Data Node :** DataNode actually stores data in the Hadoop Filesystem
- **Secondary Name Node :** perform periodic checkpoints to Name Node Metadata

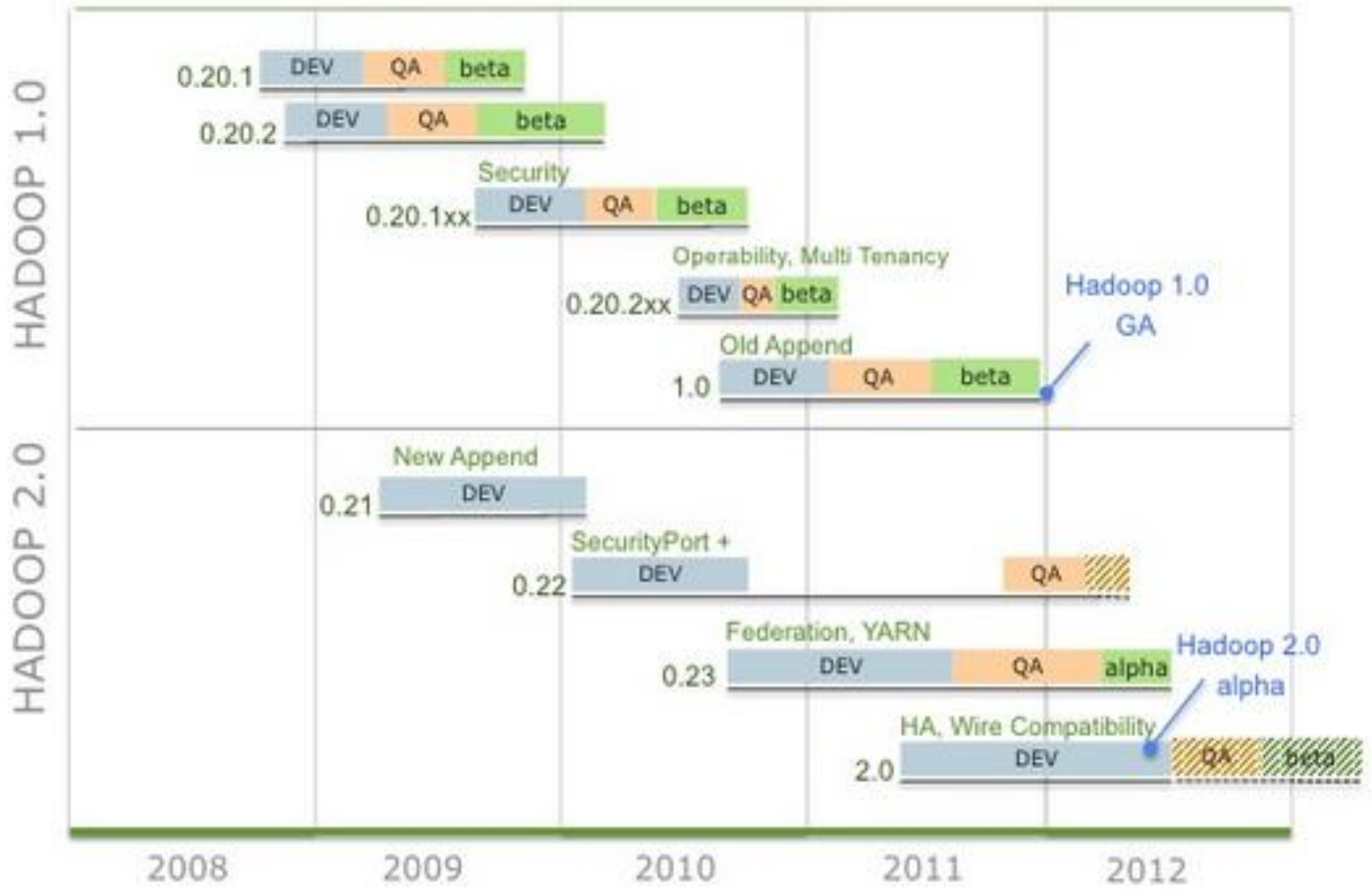


Hadoop Release

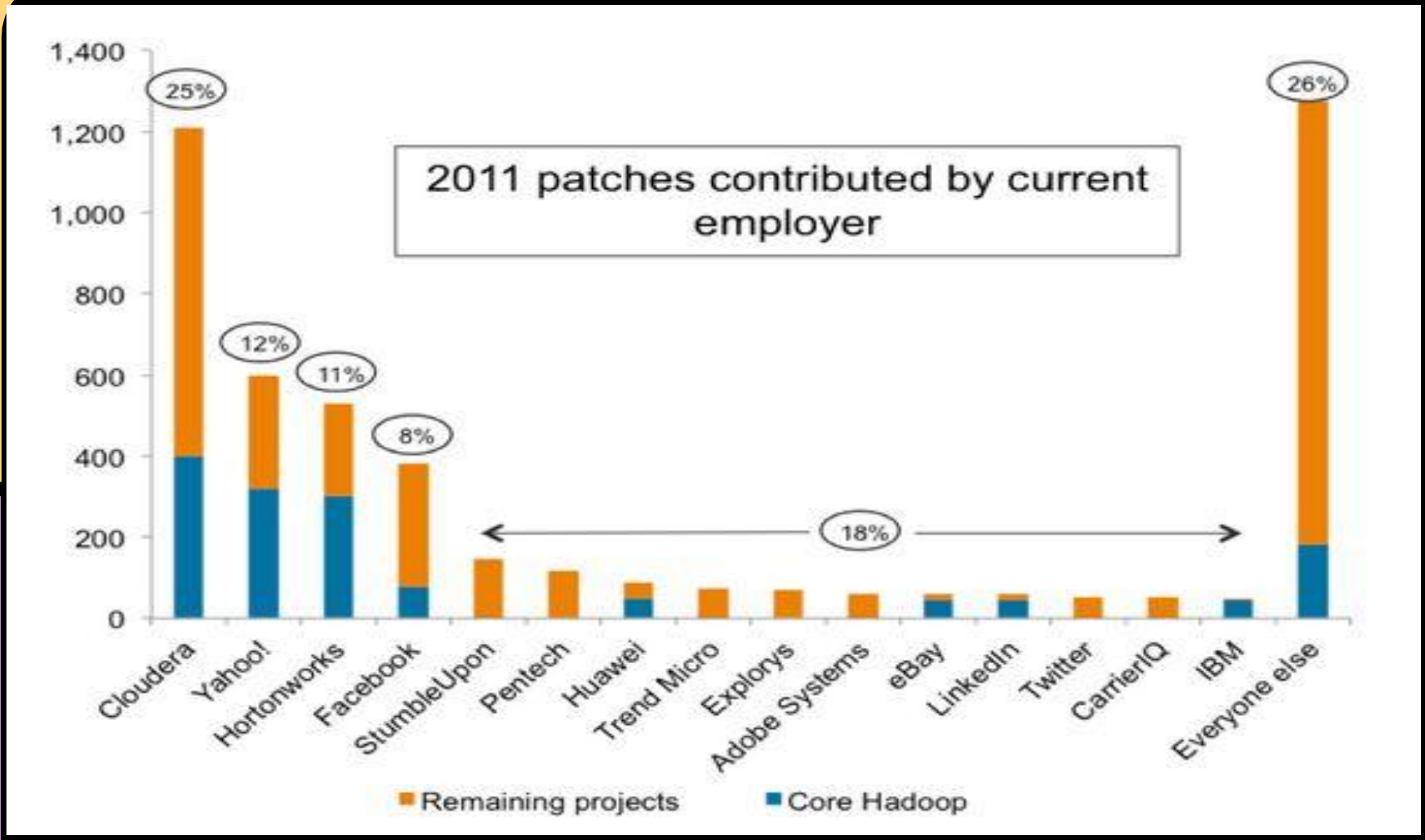
- **Available Versions**

- **1.0.X -** current stable version, 1.0 release
- **1.1.X -** current beta version, 1.1 release
- **2.X.X -** current alpha version
- **0.23.X -** simmilar to 2.X.X but missing NN HA.
- **0.22.X -** does not include security
- **0.20.203.X -** old legacy stable version
- **0.20.X -** old legacy version

Release Lifecycle



Contributors



Release Comparison

Hadoop 1 and Hadoop 2

Hadoop 1 *(GA)*

- Security
- Append/Fsync (Hbase)
- WebHdfs + Spnego
- Write pipeline improvements
- Local write optimization
- Performance improvements
- Disk-fail-in-place

Hadoop 2 (alpha)

- New Append
- Federation
- Wire compatibility
- Edit logs rewrite
- Faster startup
- HA NameNode

Limitations of MR1

- Only MR1 is supported for processing
- Single Point of failure
  - JobTracker
- Scalability limitation :
  - Max Nodes in cluster = 4000
  - Max concurrent tasks = 40000

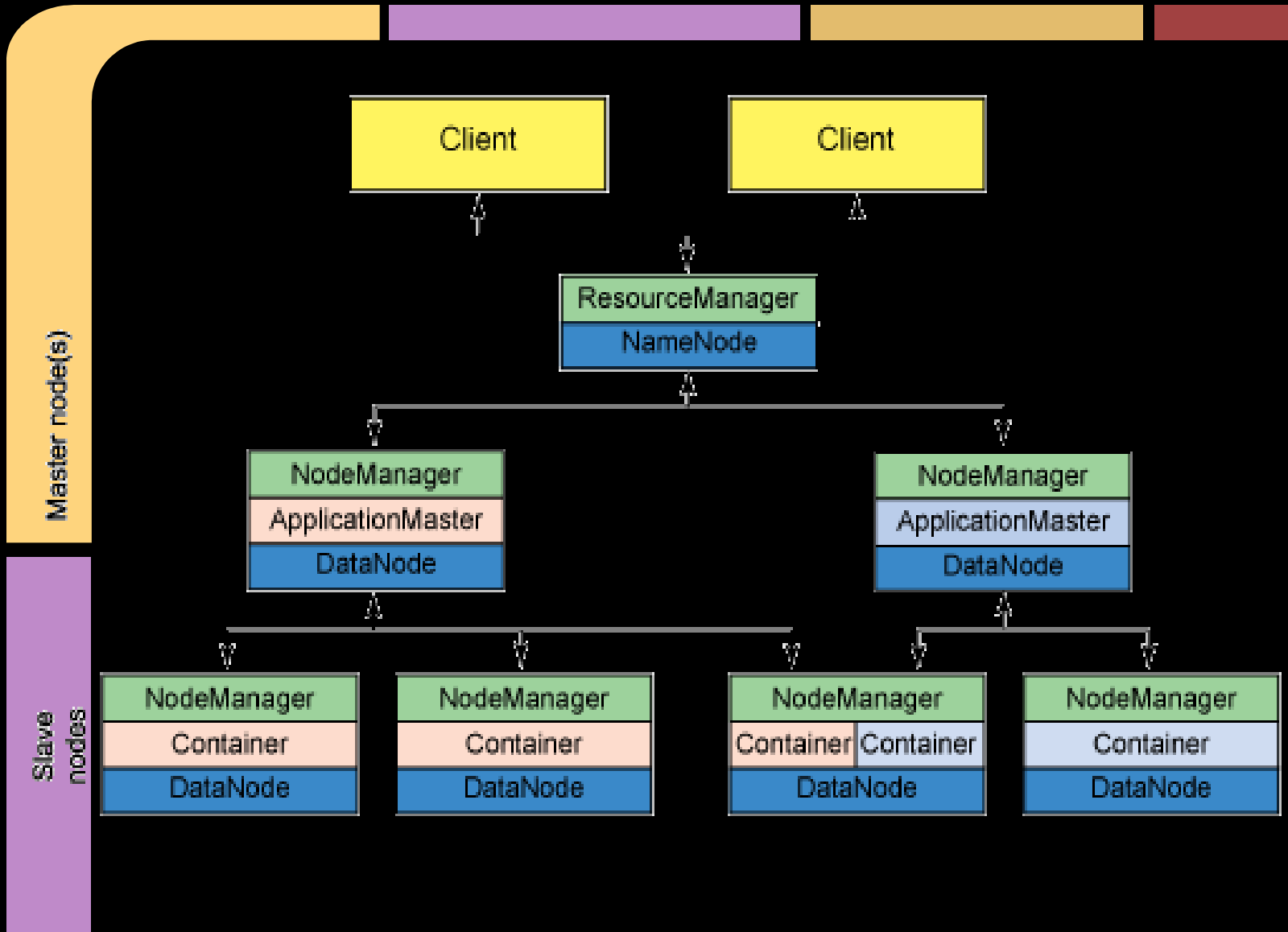
NextGen MapReduce (YARN)

- The fundamental idea of MRv2 is to split up the two major functionalities of the JobTracker, resource management and job scheduling/monitoring, into separate daemons.
- The idea is to have a global ResourceManager (*RM*) and per-application ApplicationMaster (*AM*). An application is either a single job in the classical sense of Map-Reduce jobs or a DAG of jobs.
- The ResourceManager and per-node slave, the NodeManager (*NM*), form the data-computation framework. The ResourceManager is the ultimate authority that arbitrates resources among all the applications in the system.

YARN Daemons/Components

- **Resource Manager:** It governs an entire cluster and manages the assignment of applications to underlying compute resources. RM orchestrates the division of resources (compute, memory, bandwidth, etc.) to underlying NodeManagers
- **Node Manager:** The NodeManager manages each node within a YARN cluster. It is responsible for containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager/Scheduler.
- **Scheduler :**It is responsible for allocating resources to the various running applications subject to familiar constraints of capacities, queues etc
- **Application Master:** It manages each instance of an application that runs within YARN. The ApplicationMaster is responsible for negotiating resources from the ResourceManager and, through the NodeManager, monitoring the execution and resource consumption of containers (resource allocations of CPU, memory, etc.)
- **Container** : It represents an allocated resource in the cluster.

YARN Architecture



Single Node Cluster Configuration

- **Prerequisites:**
  - System: Mac OS / Linux
  - Java 6 installed
  - Dedicated user for hadoop
  - SSH configured

Single Node Cluster Configuration

Steps to configure:

1)Download Hadoop release from hadoop release page
http://hadoop.apache.org/releases.html

2)Extract the tar ball

3)Setup environment with adding variables :

HADOOP_HOME, HADOOP_MAPRED_HOME, HADOOP_COMMON_HOME, YARN_HOME, HADOOP_HDFS_HOME, HADOOP_CONF_DIR

4)Create two directories to be used by NameNode and DataNode

Configuration Files

- Main configuration files that need to be configured:
  etc/hadoop/hadoop-env.sh
  etc/hadoop/yarn-site.xml
  etc/hadoop/core-site.xml
  etc/hadoop/hdfs-site.xml
  etc/hadoop/mapred-site.xml

Single Node Cluster Configuration

- Format Name Node filesystem:

  $ bin/hadoop namenode –format
- Start the hadoop daemons:
  $ sbin/start-all.sh
- Check the running java daemons:

  $jps
- When you're done, stop the daemons with:
  $ sbin/stop-all.sh

Web Interfaces

- HDFS web interface:
  http://localhost:50070
- All Application interface:
  http://localhost:8088

References

- http://hadoop.apache.org/
- http://wiki.apache.org/hadoop/
- http://hadoop.apache.org/core/docs/current/hdfs_design.html
- http://wiki.apache.org/hadoop/FAQ