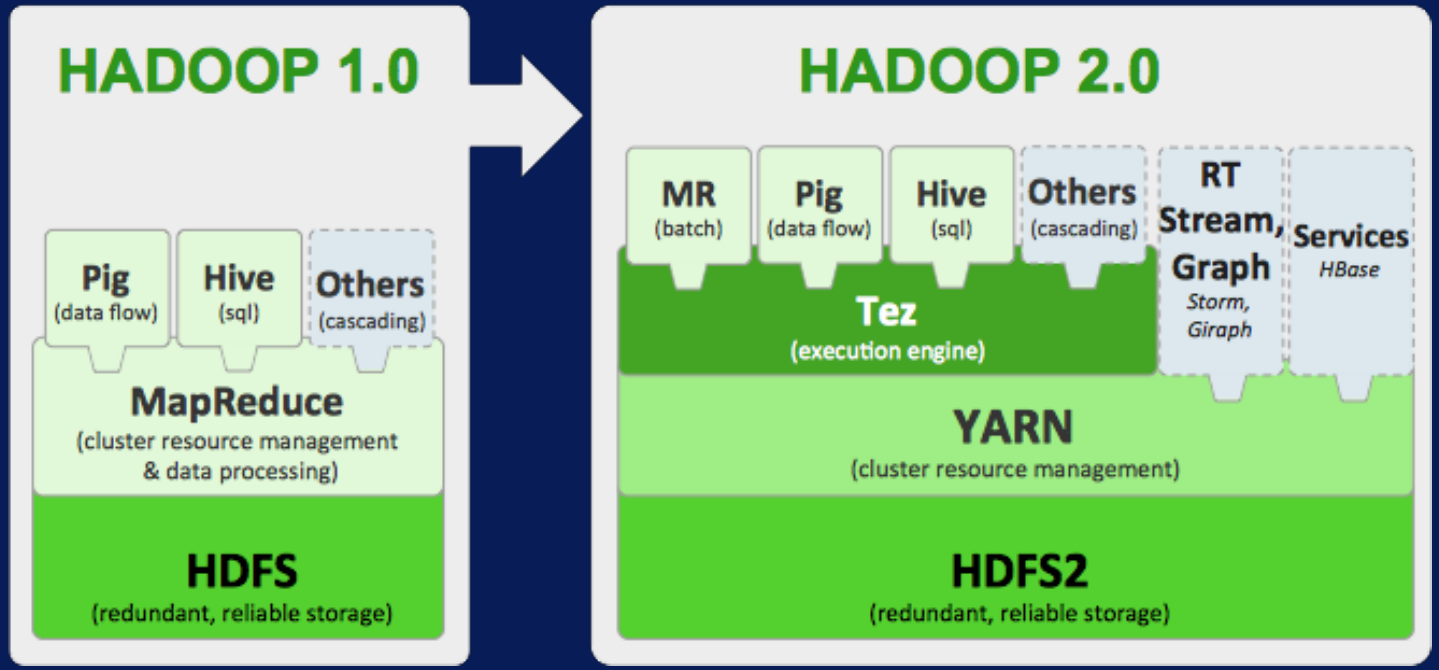
The Basic Hadoop Components

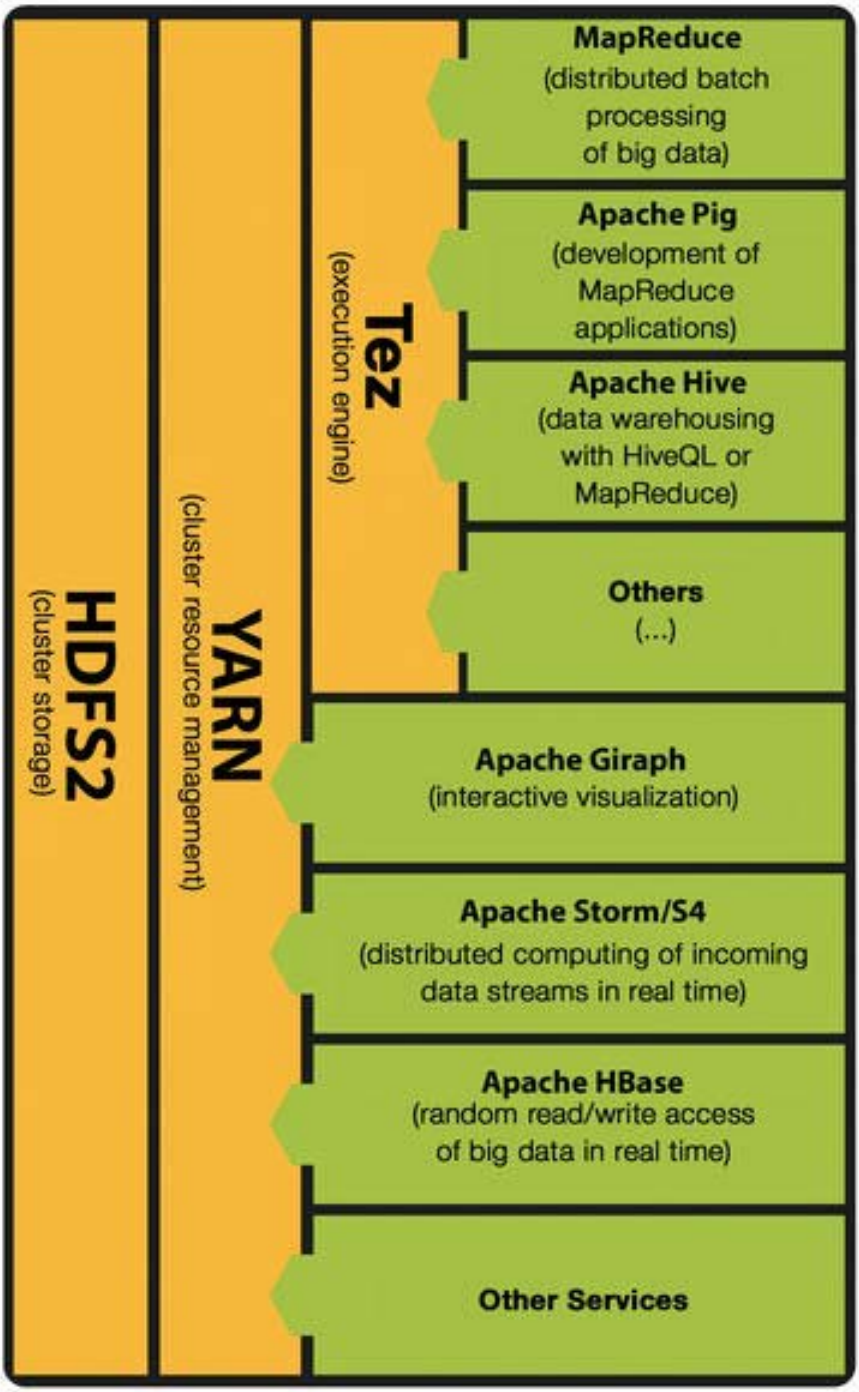
- ***Hadoop Common*** **- libraries and utilities**
- ***Hadoop Distributed File System (HDFS)*** **– a distributed file-system**
- ***Hadoop YARN*** **– a resource-management platform, scheduling**
- ***Hadoop MapReduce*** **– a programming model for large scale data processing**

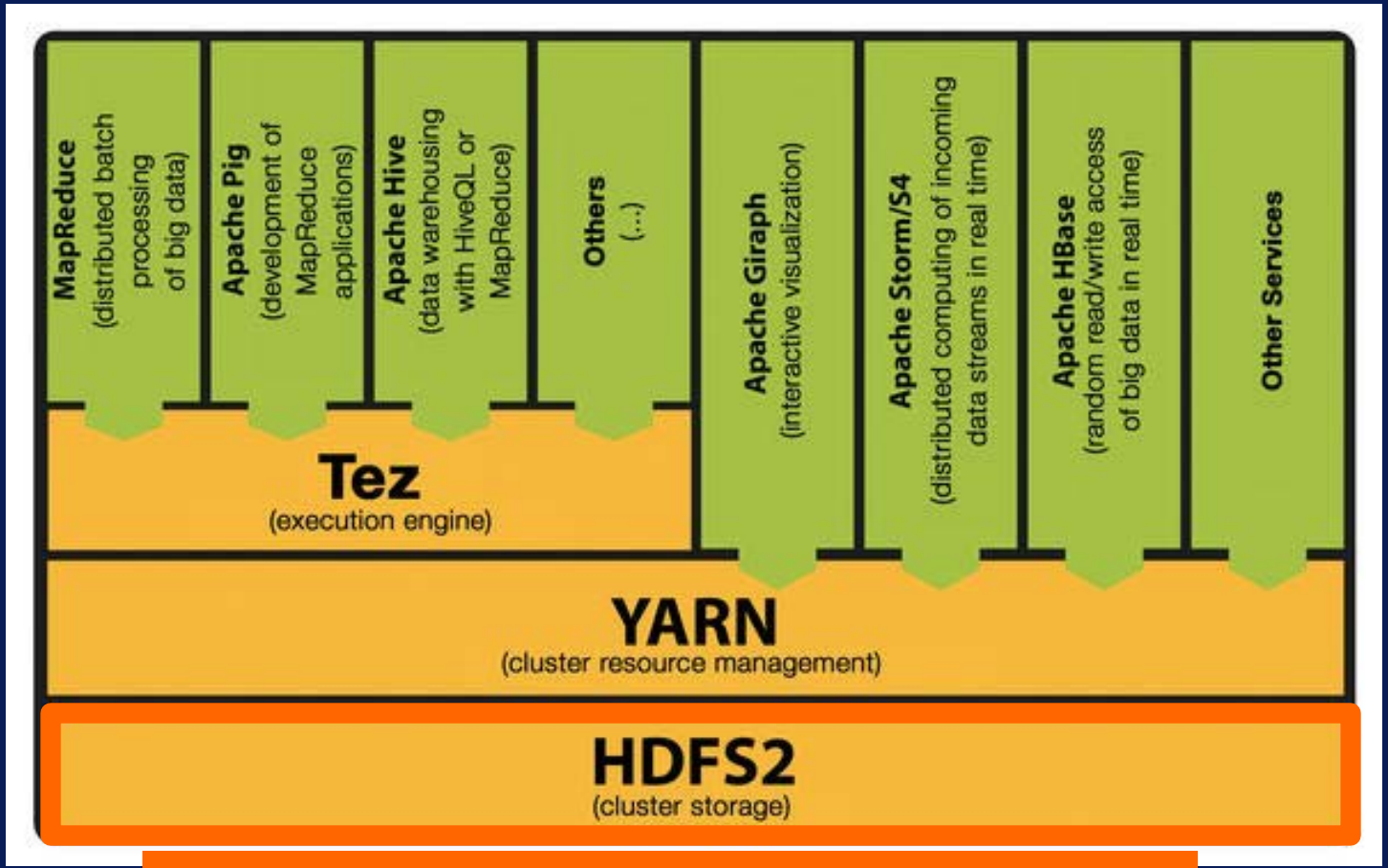
Hadoop Stack Transition



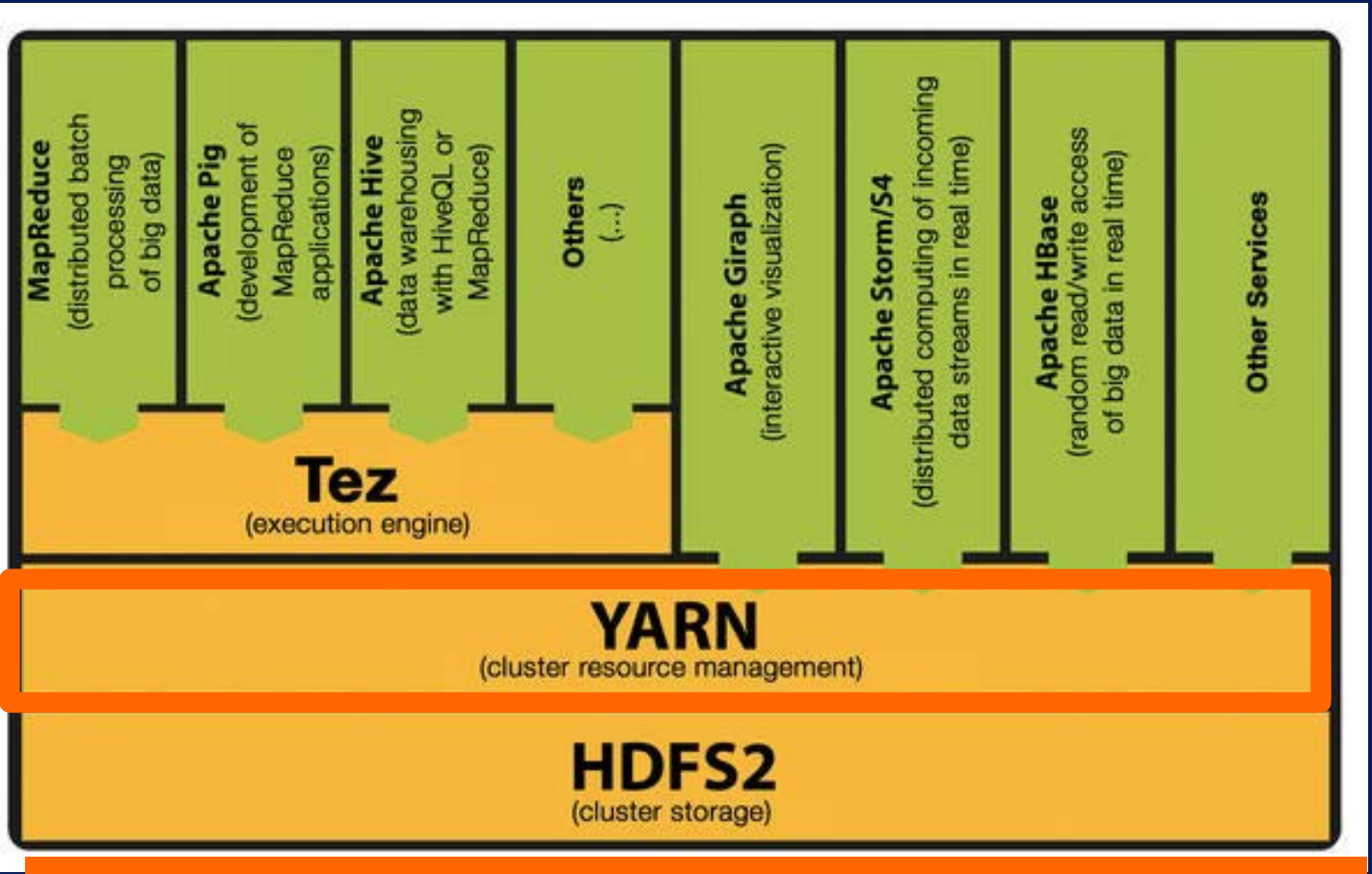
Applications and Frameworks

- ***HBase*** – a scalable data warehouse with support for large tables.
- ***Hive*** – a data warehouse infrastructure that provides data summarization and ad hoc querying
- ***Pig*** – A high-level data-flow language and execution framework for parallel computation
- ***Spark*** – a fast and general compute engine for Hadoop data. Wide range of applications – ETL, Machine Learning, stream processing, and graph analytics.

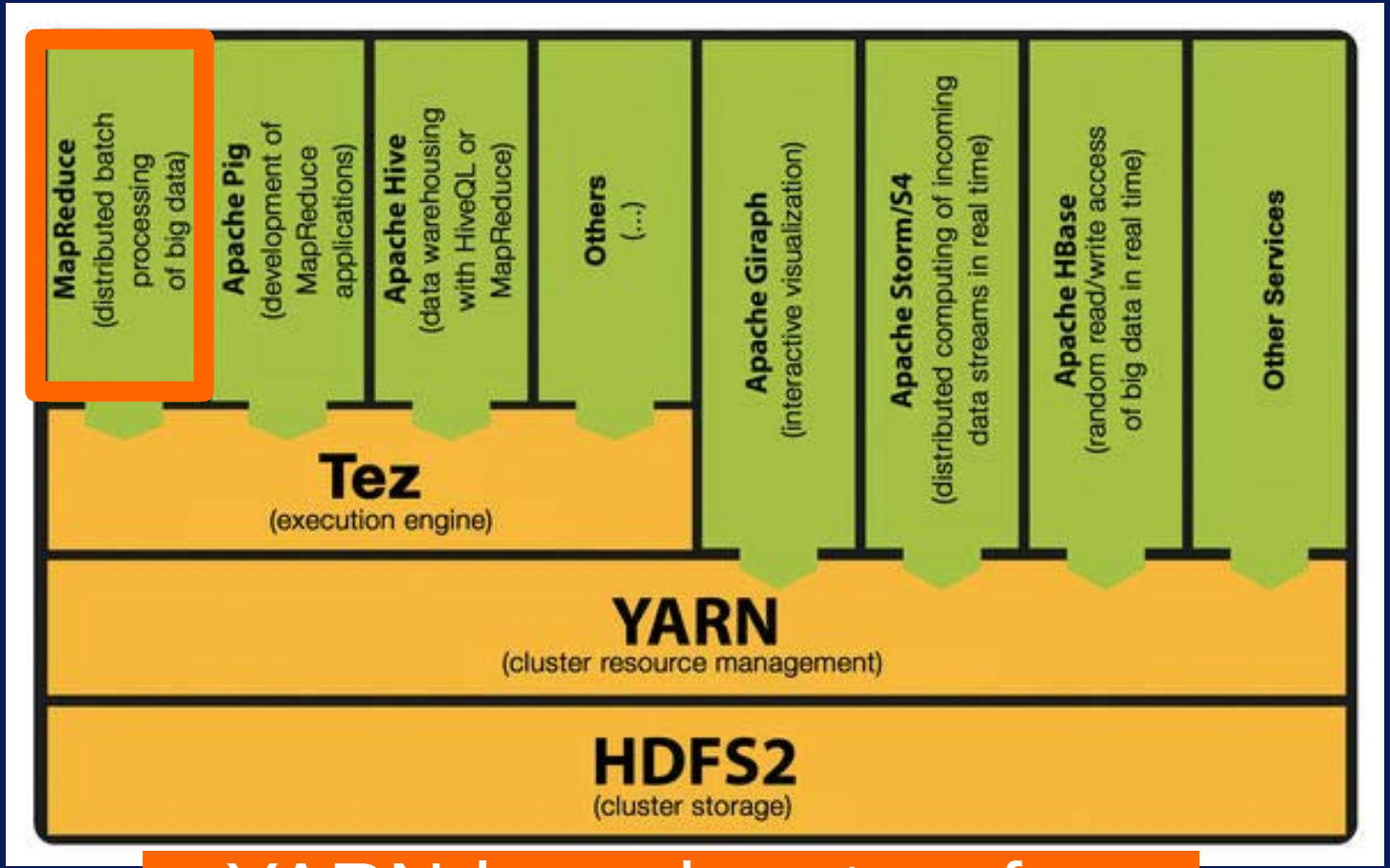

HDFS2
(cluster storage)
YARN
(cluster resource management)
Tez
(execution engine)
MapReduce
(distributed batch processing of big data)
Apache Pig
(development of MapReduce applications)
Apache Hive
(data warehousing with HiveQL or MapReduce)
Others
(...)
Apache Giraph
(interactive visualization)
Apache Storm/S4
(distributed computing of incoming data streams in real time)
Apache HBase
(random read/write access of big data in real time)
Other Services



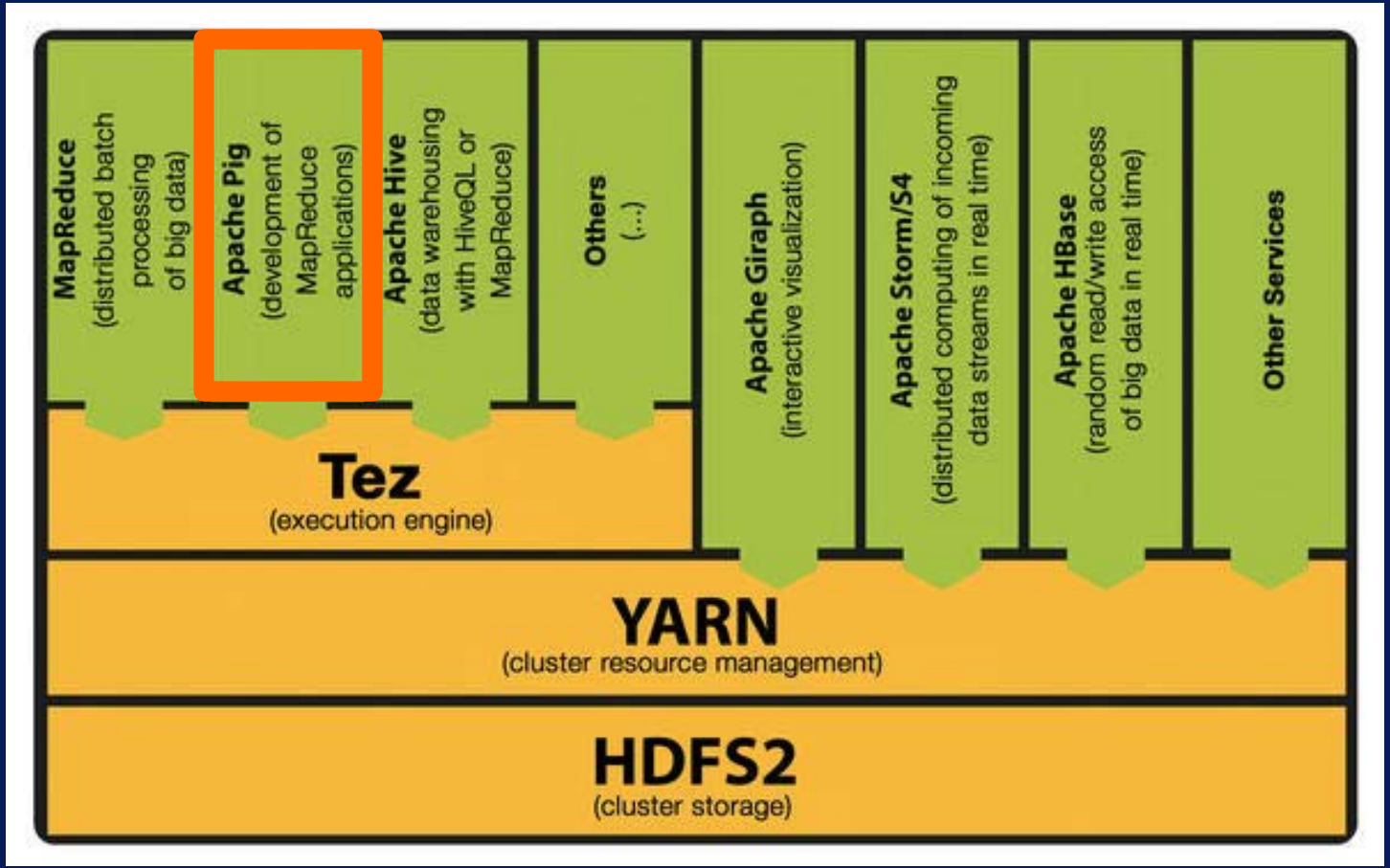
Distributed Filesystem



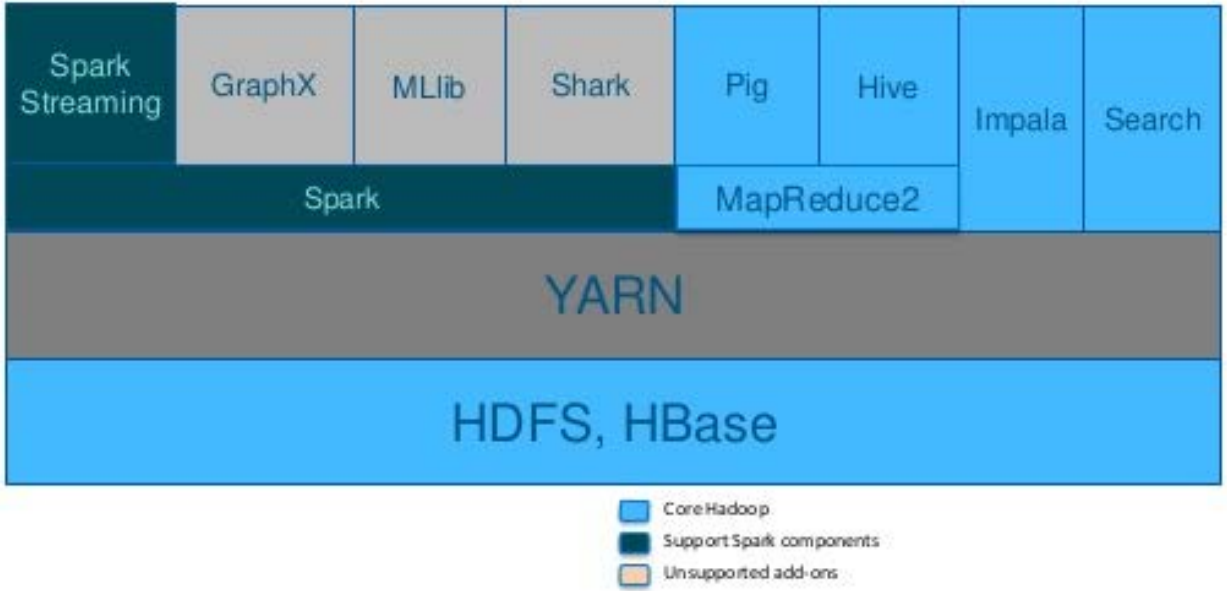
Resource Management, Scheduling



YARN-based system for parallel processing of data

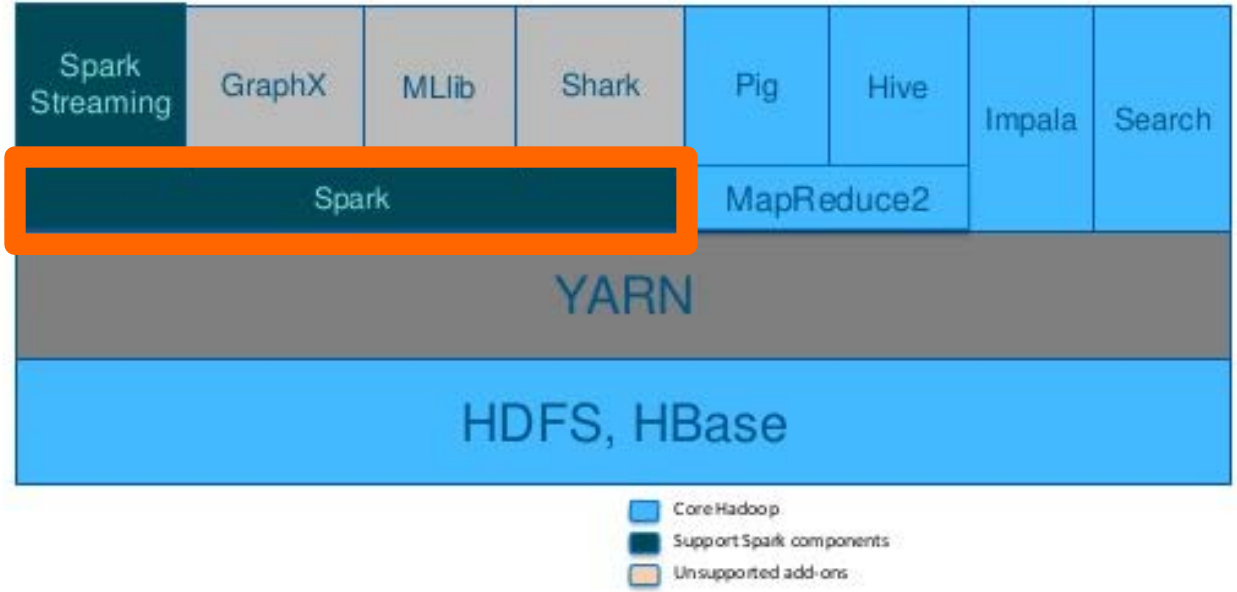

MapReduce
(distributed batch processing of big data)
Apache Pig
(development of MapReduce applications)
Apache Hive
(data warehousing with HiveQL or MapReduce)
Others
(...)
Apache Giraph
(interactive visualization)
Apache Storm/S4
(distributed computing of incoming data streams in real time)
Apache HBase
(random read/write access of big data in real time)
Other Services
Tez
(execution engine)
YARN
(cluster resource management)
HDFS2
(cluster storage)

Hadoop in the Spark world



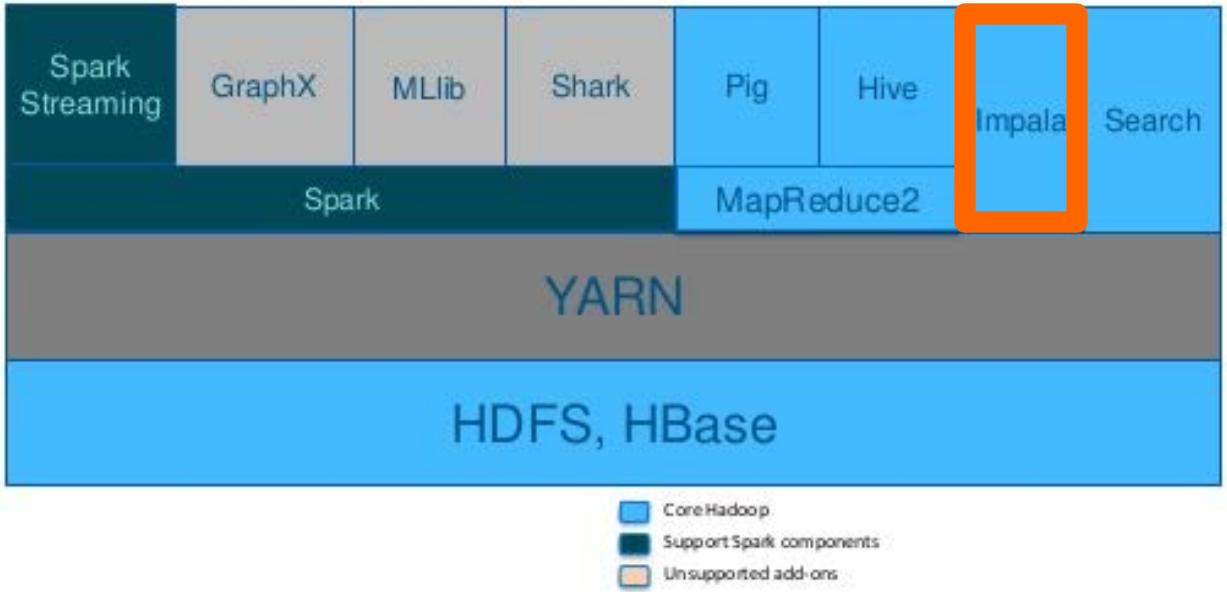
cloudera

Hadoop in the Spark world



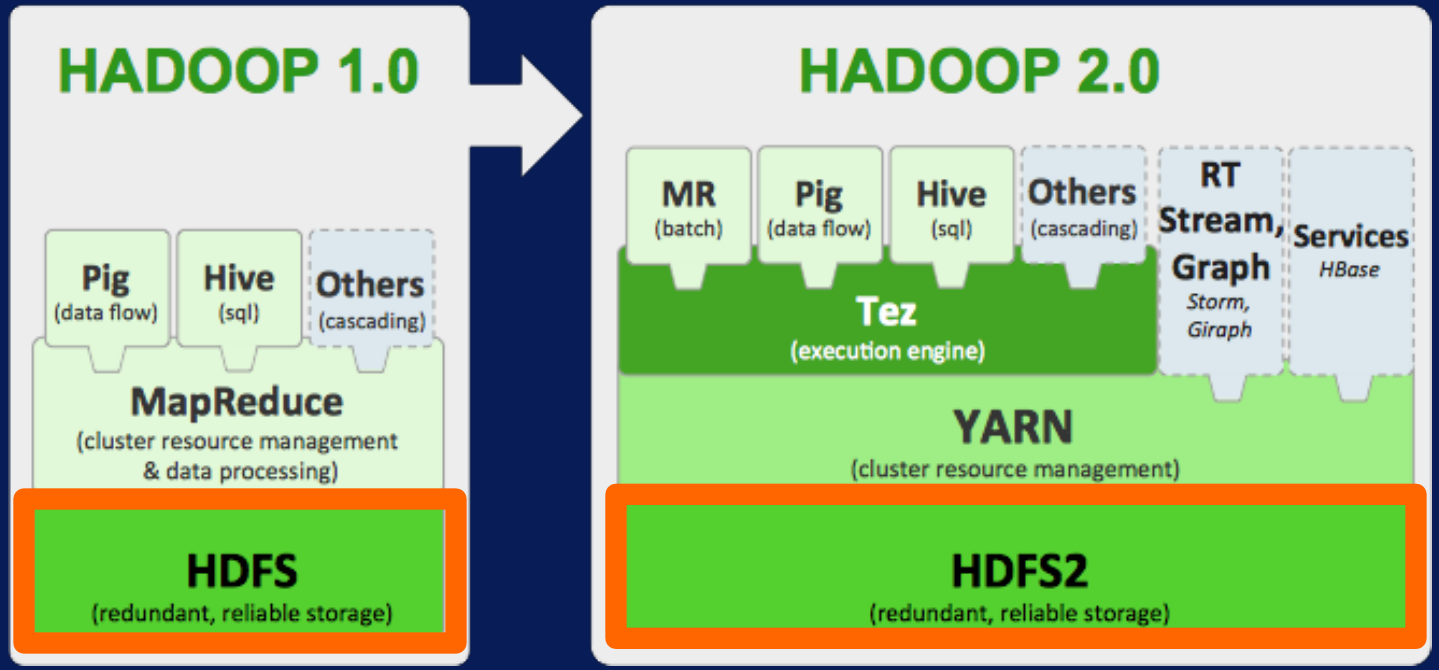
cloudera

Hadoop in the Spark world



cloudera

HDFS and HDFS2



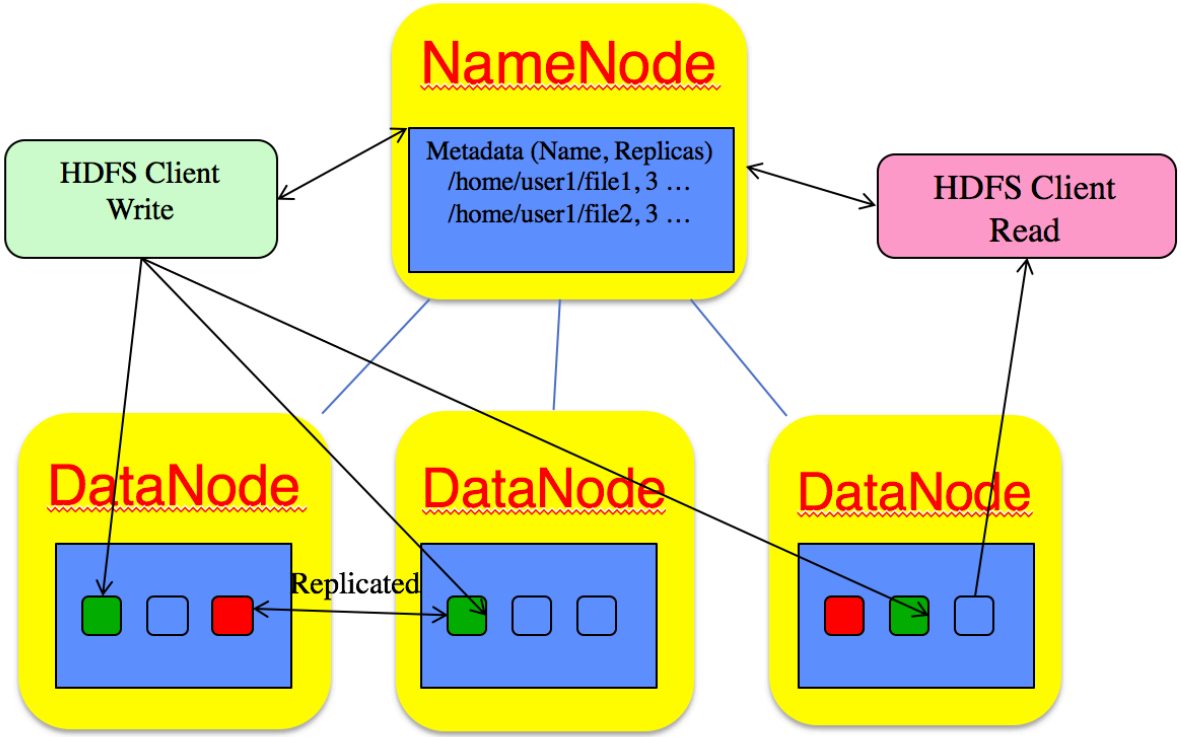
Original HDFS Design Goals

- ***Resilience to hardware failure***
- ***Streaming data access***
- ***Support for large dataset, scalability to hundreds/thousands of nodes with high aggregate bandwidth***
- ***Application locality to data***
- ***Portability across heterogeneous hardware and software platforms***

Original HDFS Design

- ***Single NameNode -*** a master server that manages the file system namespace and regulates access to files by clients.
- ***Multiple DataNodes –*** typically one per node in the cluster. Functions:
  - Manage storage
  - Serving read/write requests from clients
  - Block creation, deletion, replication

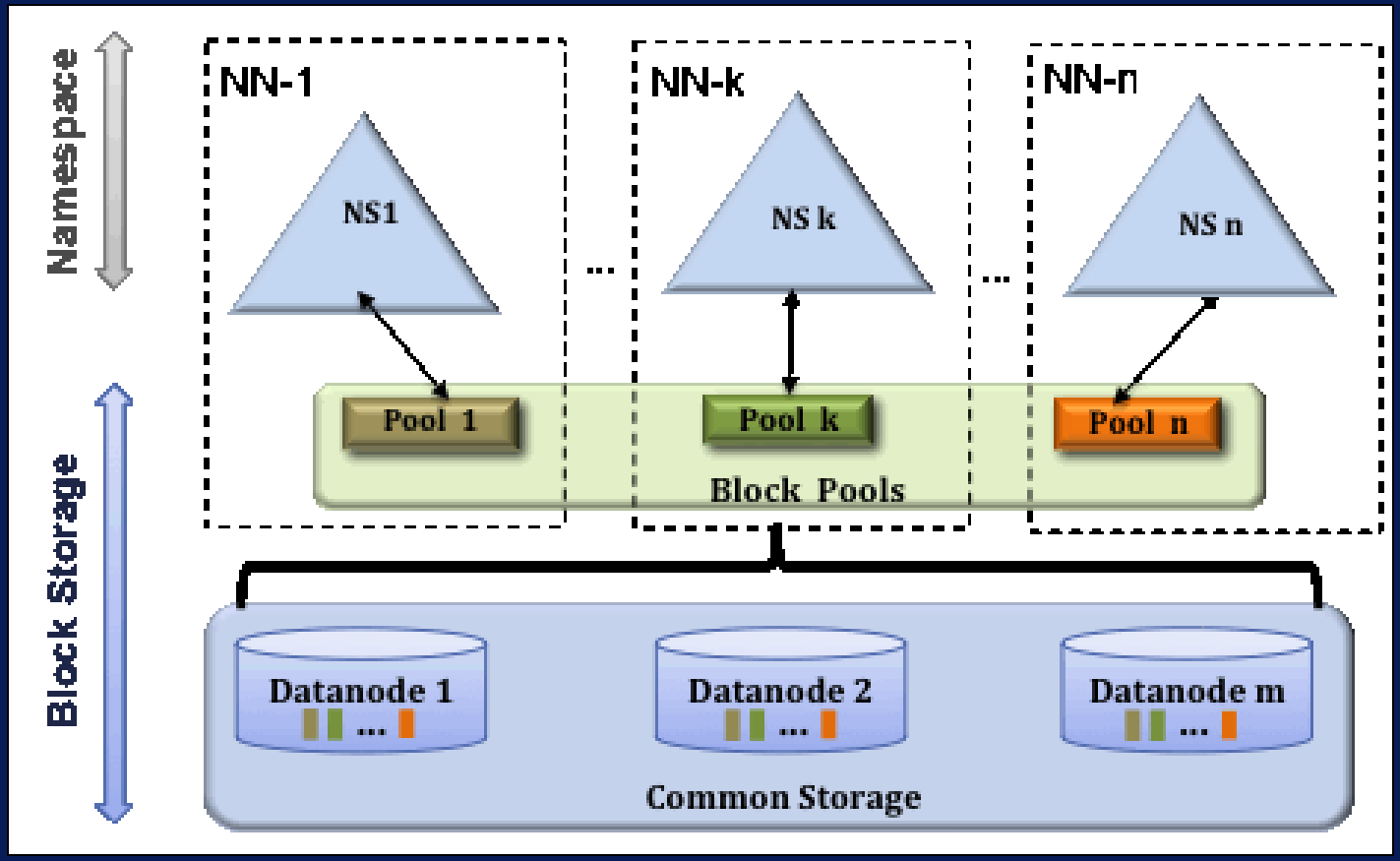
    based on instructions from NameNode


NameNode
Metadata (Name, Replicas)
/home/user1/file1, 3 …
/home/user1/file2, 3 …
HDFS Client
Write
HDFS Client
Read
DataNode
DataNode
DataNode
Replicated

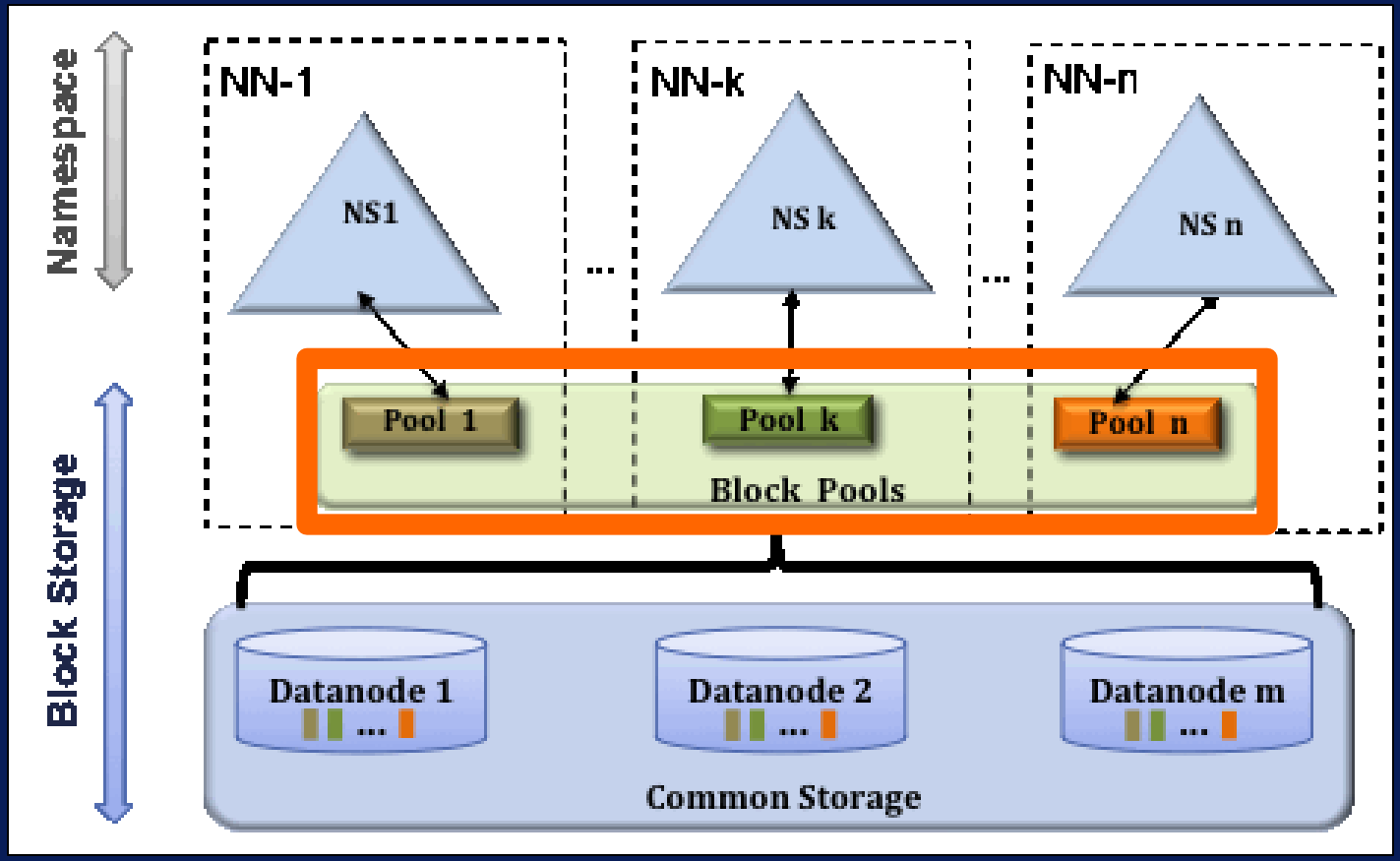
HDFS in Hadoop 2

- ***HDFS Federation***
- ***Multiple Namenode servers***
- ***Multiple namespaces***
- ***High Availability – redundant NameNodes***
- ***Heterogeneous Storage and Archival Storage***
  - *ARCHIVE, DISK, SSD, RAM_DISK*

Federation


Namespace
Block Storage
NN-1
NN-k
NN-n
NS1
NS k
NS n
Pool 1
Pool k
Pool n
Block Pools
Datanode 1
Datanode 2
Datanode m
Common Storage

Federation: Block Pools



Federation: Benefits

- ***Allows namespace scaling***
- ***Scales up filesystem read/write throughput***
- ***Isolation***

MapReduce Framework

- ***Software framework* – *for writing parallel data processing applications***
- ***MapReduce job splits data into chunks***
- ***Map tasks process data chunks***
- ***Framework sorts map output***
- ***Reduce tasks use sorted map data as input***

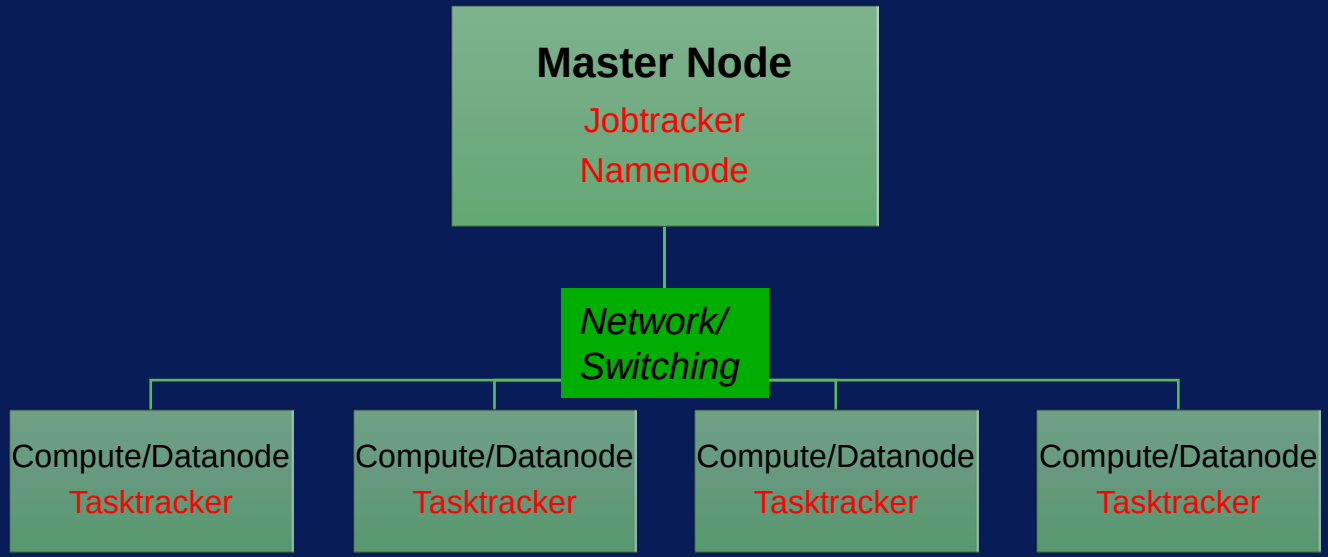
MapReduce Framework

- *Typically compute and storage nodes are the same.*
- *MapReduce tasks and HDFS running on the same nodes*
- *Can schedule tasks on nodes with data already present.*

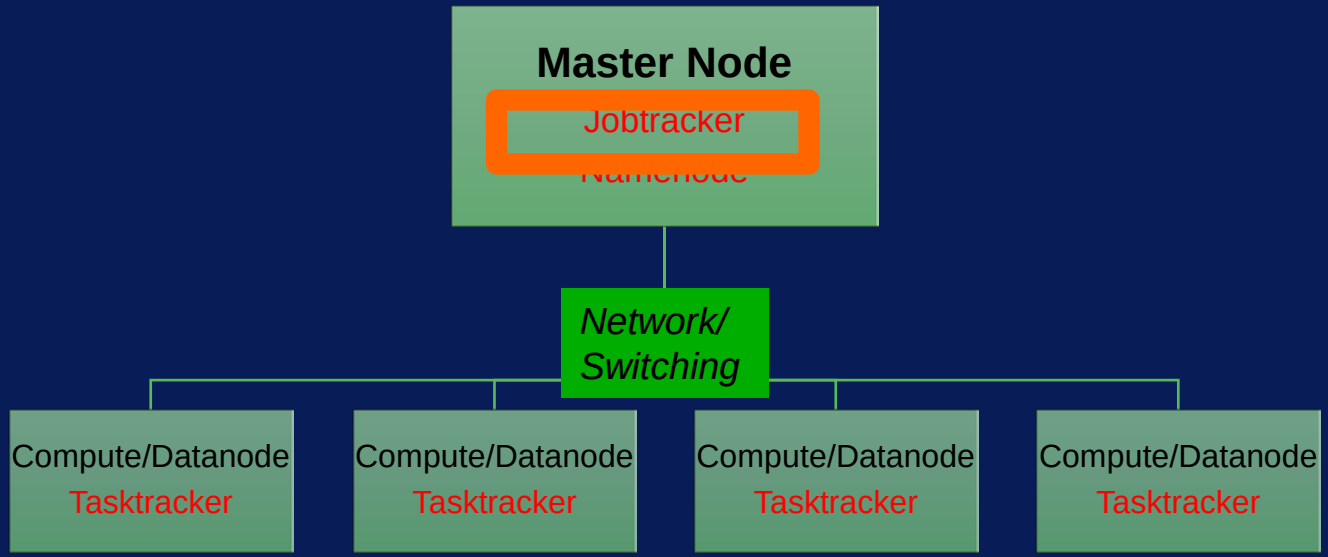
Original MapReduce Framework

- ***Single master JobTracker***
- ***JobTracker schedules, monitors, and re-executes failed tasks.***
- ***One slave TaskTracker per cluster node***
- ***TaskTracker executes tasks per JobTracker requests.***

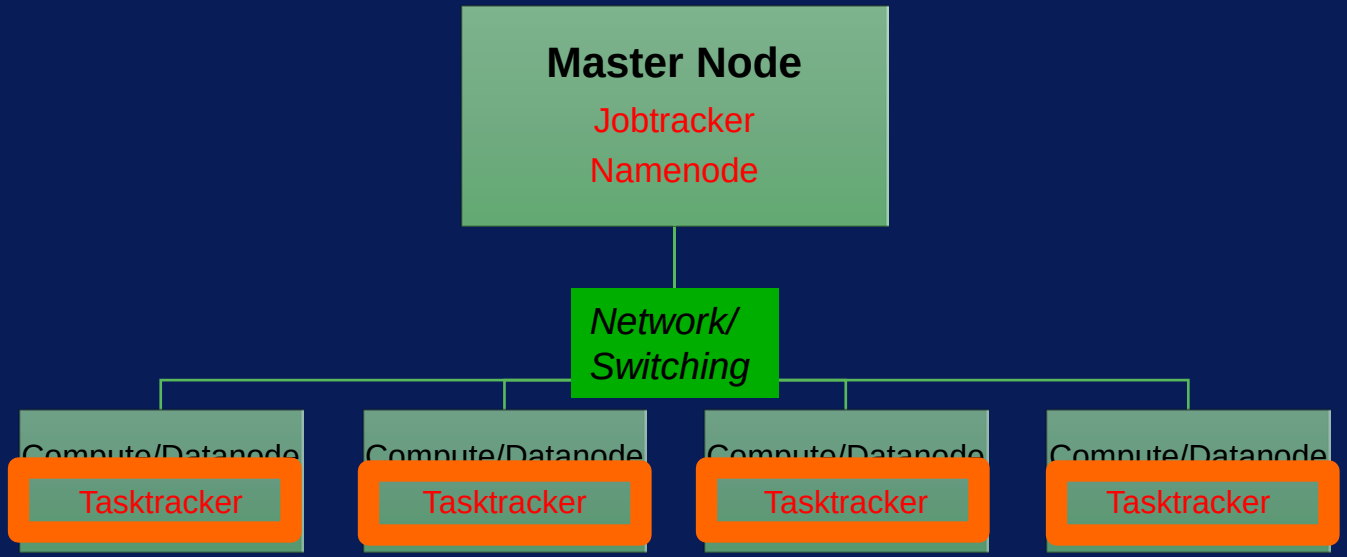
Original Hadoop Architecture



Original Hadoop Architecture



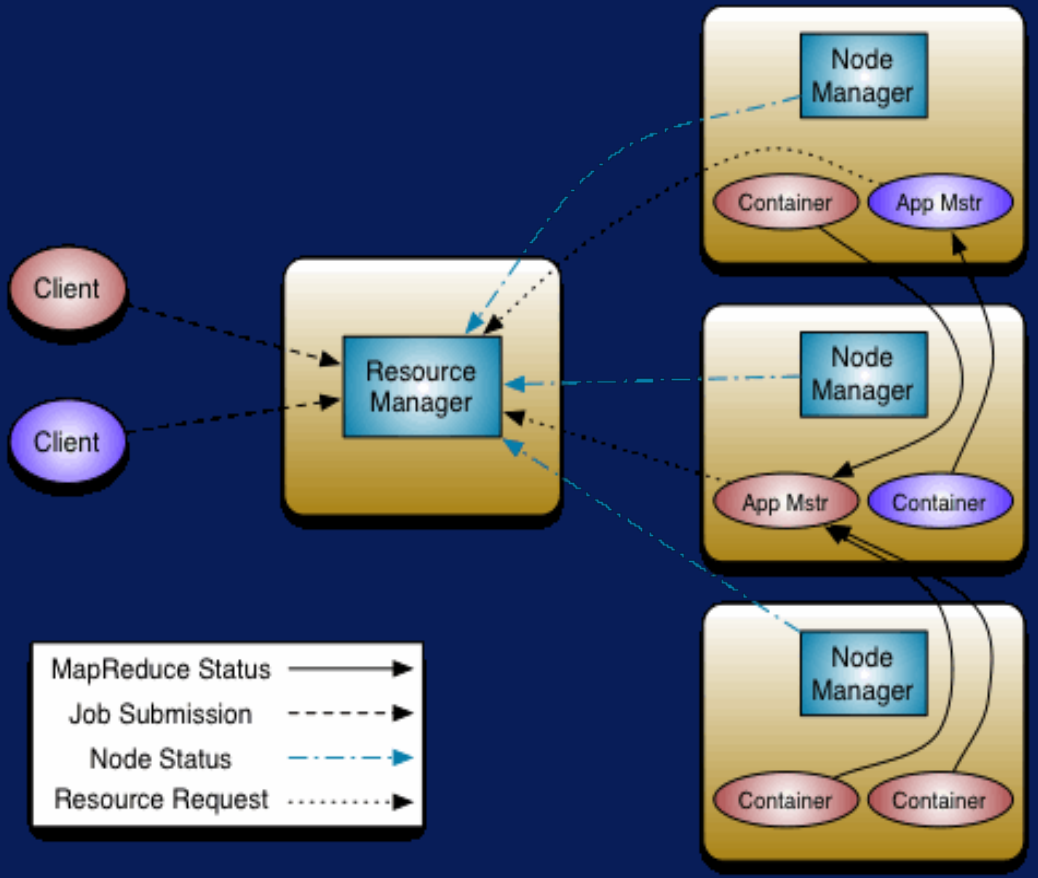
Original Hadoop Architecture



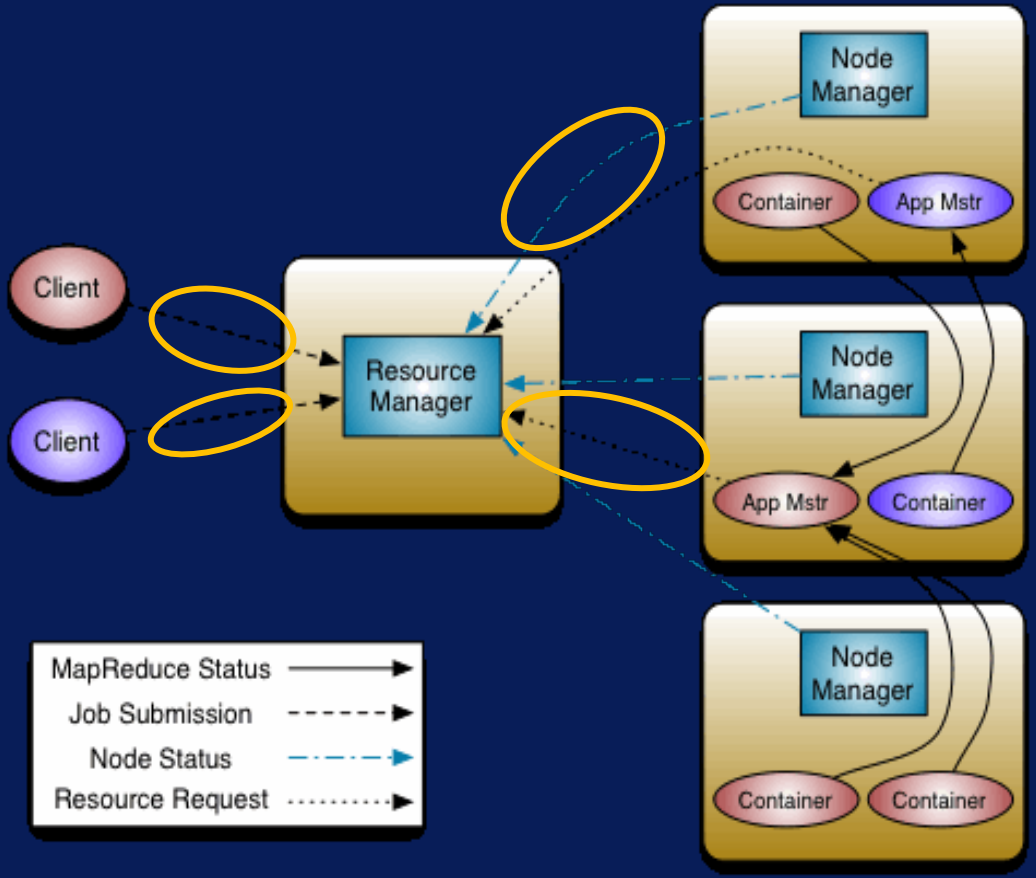
YARN: NexGen MapReduce

- ***Main idea – Separate resource management and job scheduling/monitoring.***
- ***Global ResourceManager (RM)***
- ***NodeManager on each node***
- ***ApplicationMaster – one for each application***

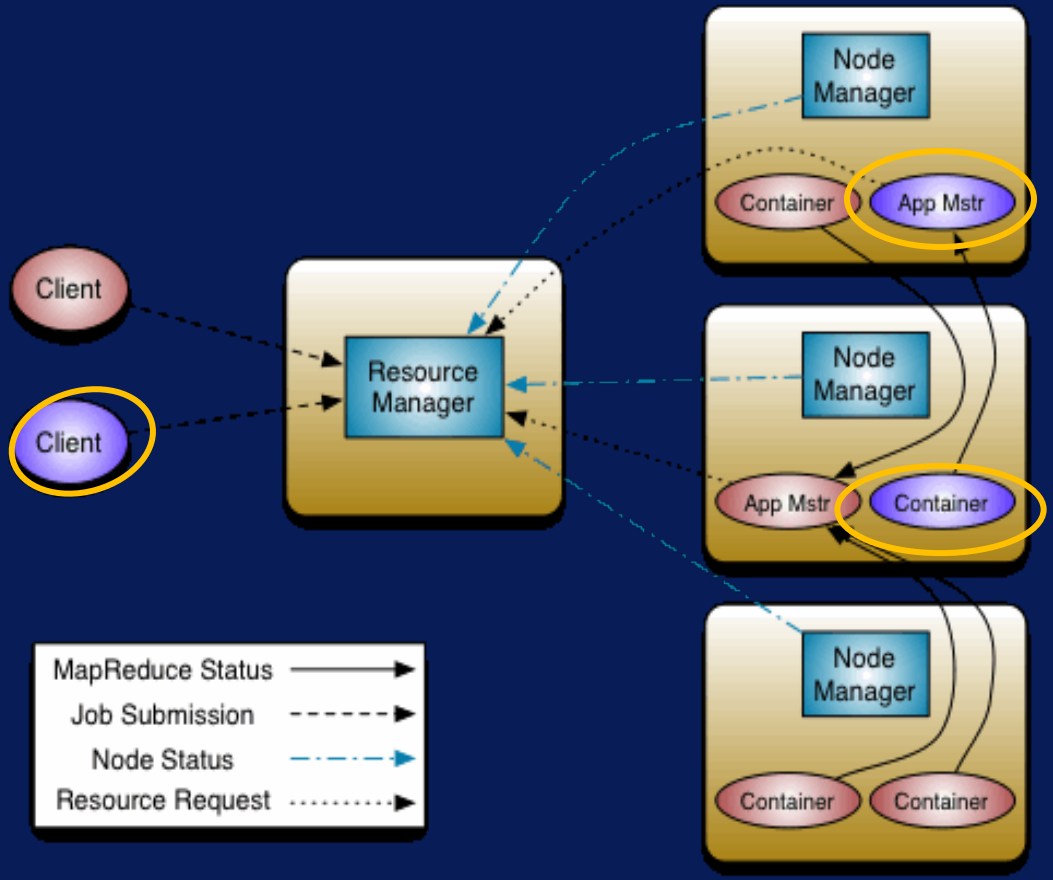
YARN Architecture



YARN Architecture



YARN Architecture



Additional YARN Features

- *High Availability ResourceManager*
- *Timeline Server*
- *Use of Cgroups*
- *Secure Containers*
- *YARN – web services*
  REST APIs