

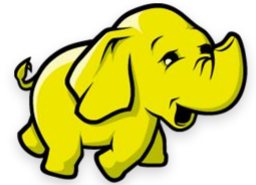


Data Engineering: MapReduce, Hadoop and Spark

Apache Hadoop

Apache Hadoop is an open source software framework for storage and large scale processing of data-sets on clusters of commodity hardware.

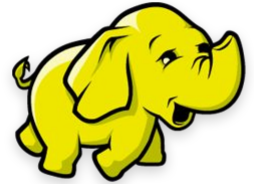
- Created by Doug Cutting and Mike Cafarella in 2005
- Named the project after son's toy elephant



Data Engineering:
MapReduce, Hadoop and spark

Apache Hadoop: Inspired by Google's

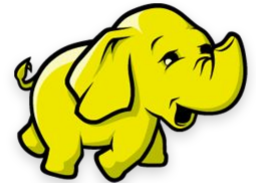
- GFS (2003)
- MapReduce (2004)
- BigTable (2006)



Data Engineering:
MapReduce, Hadoop and spark

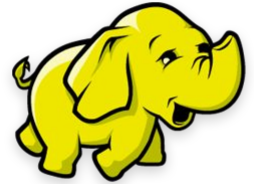
Size Timeline

| Date | Size reported by Yahoo |
|---------------|---------------------------|
| April 2006 | 188 |
| May 2006 | 300 |
| October 2006 | 600 |
| April 2007 | 1,000 |
| February 2008 | 10,000 (index generation) |
| March 2009 | 24,000 (17 clusters) |
| June 2011 | 42,000 (100+ PB) |



Apache Framework Basic Modules

- Hadoop Common
- Hadoop Distributed File System (HDFS)
- Hadoop Yarn
- Hadoop MapReduce



Hadoop 1.x VS Hadoop 2.x

Hadoop 1.x

MapReduce

(Cluster Resource management
& data processing)

HDFS

(redundant, reliable storage)

Hadoop 2.x

MapReduce

(Data processing)

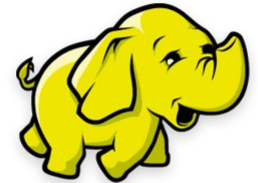
Others

YARN

(cluster resource management)

HDFS2

(redundant, highly-available & reliable storage)

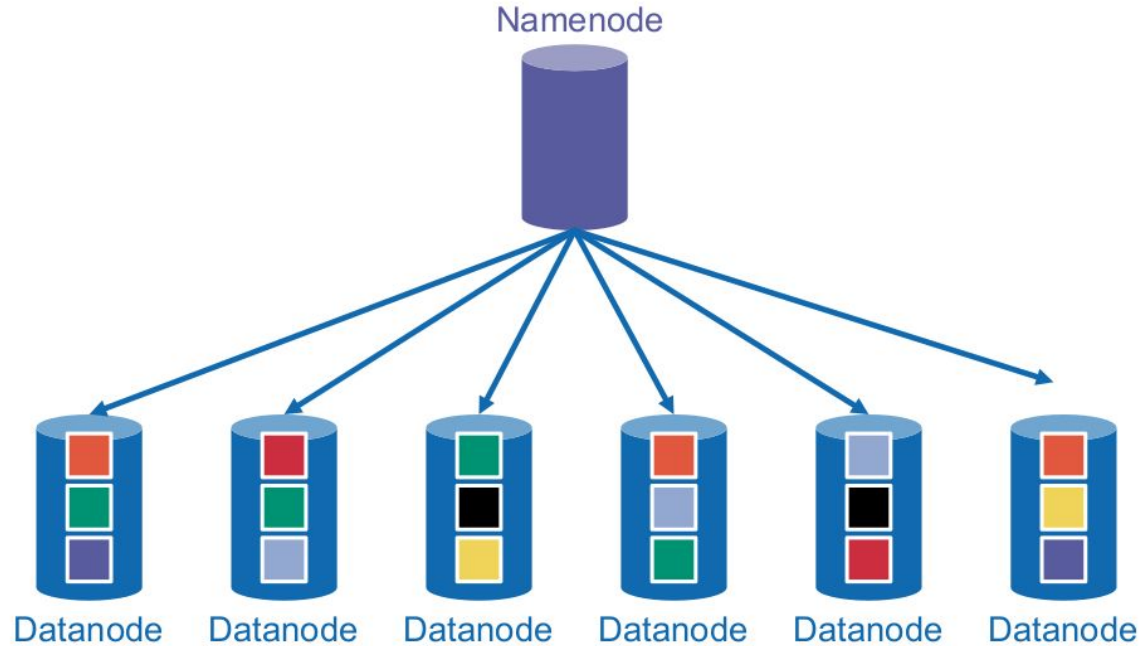


Hadoop Distributed File System (HDFS)

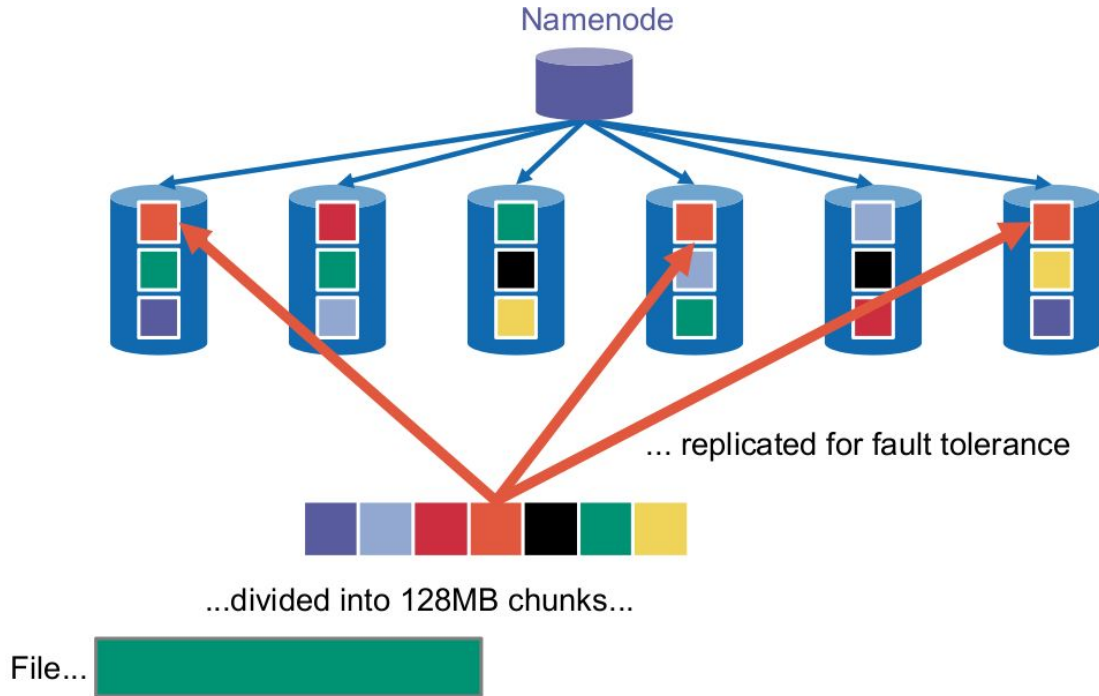
- Hardware Failure
- Batch Processing
- Large Datasets
- Simple Coherency model



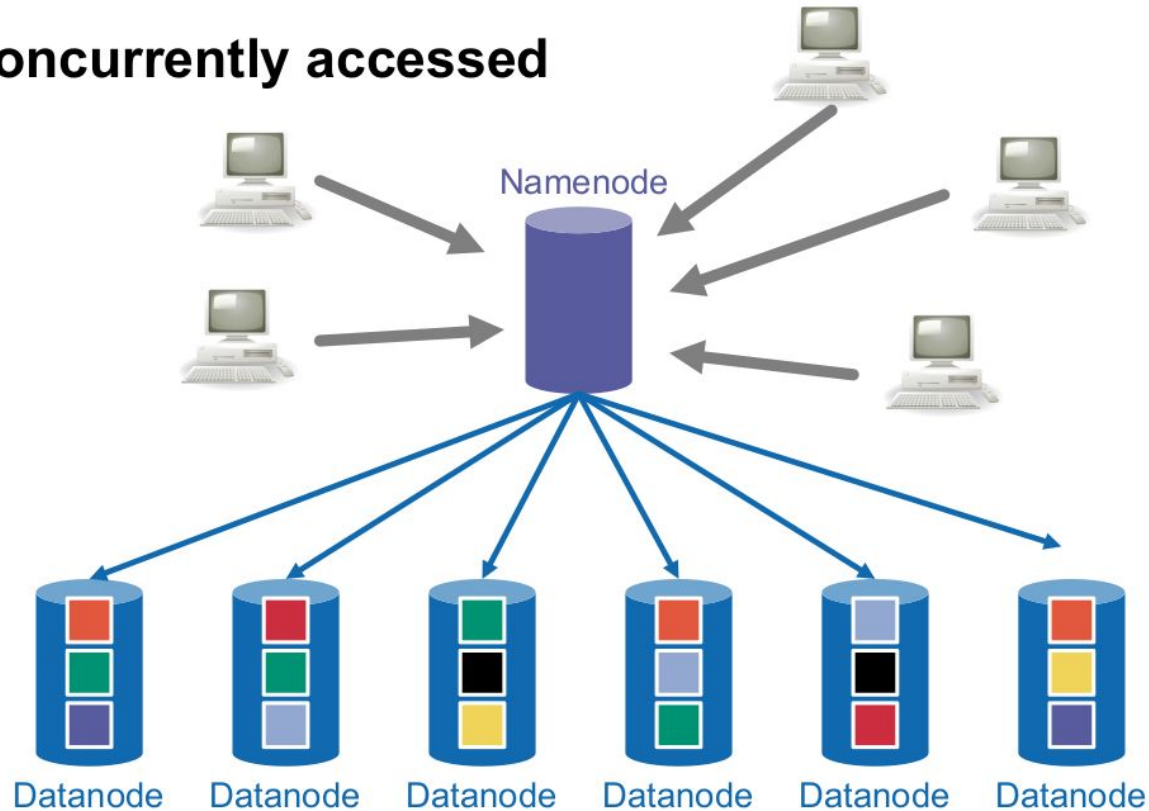
HDFS Server Architecture



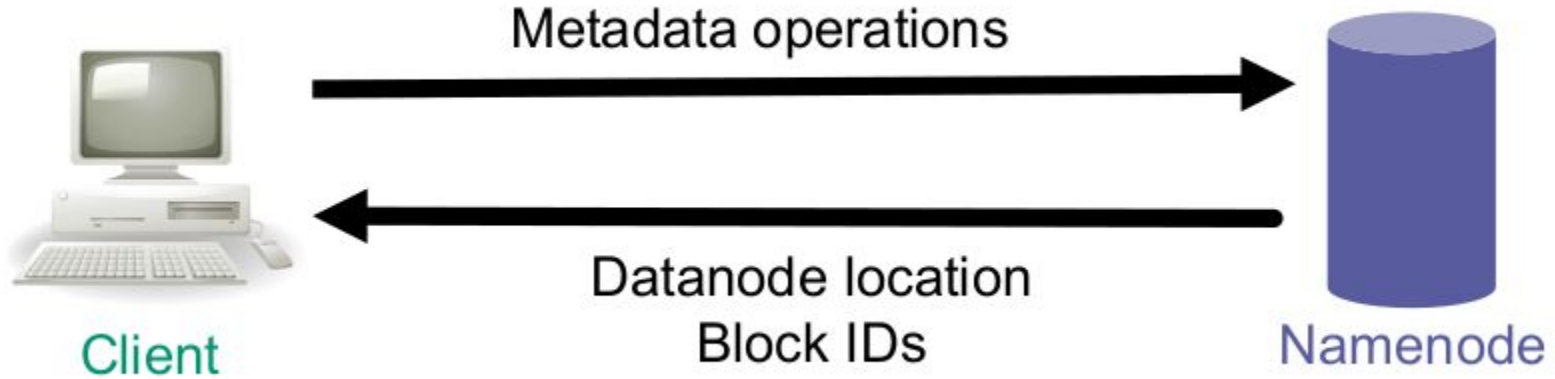
From the File Perspective



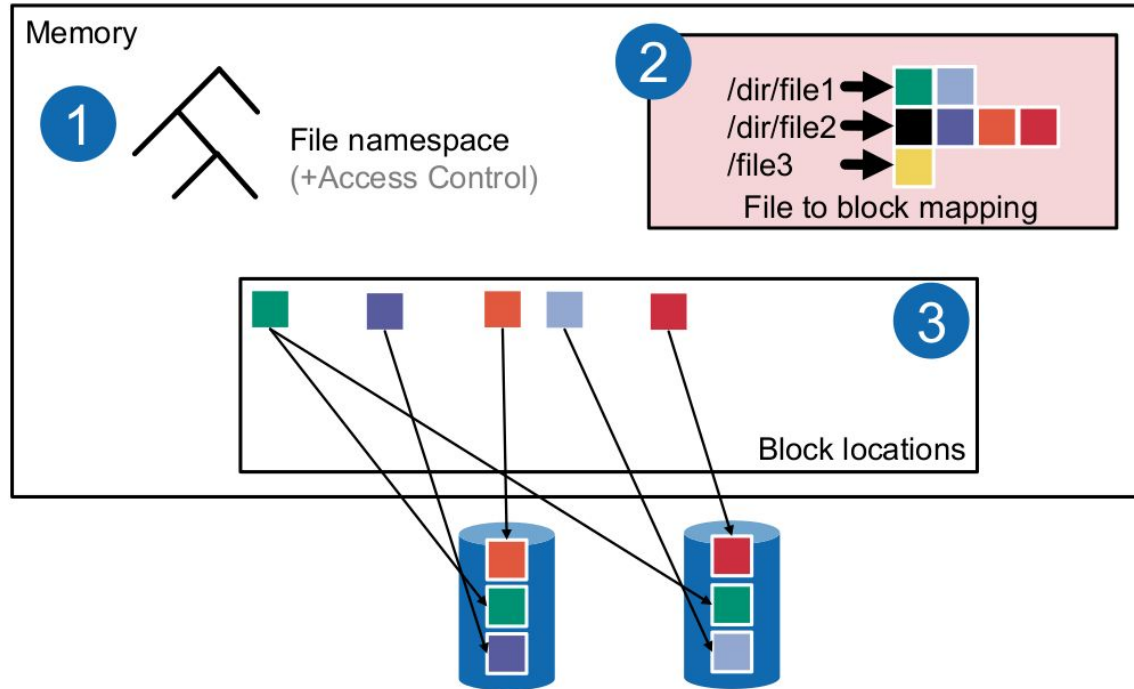
Concurrently accessed



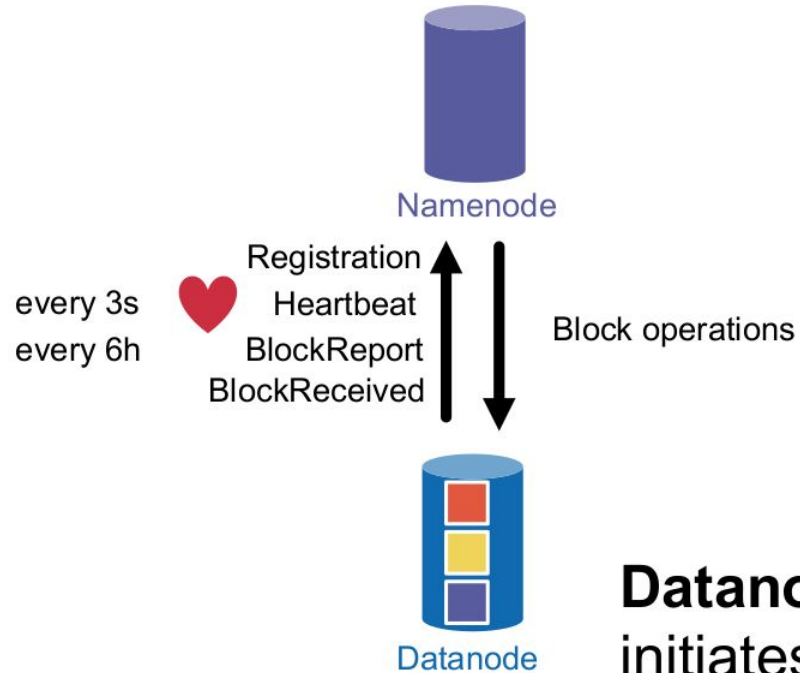
Client Protocol



NameNode: All System-wide Activity



DataNode Protocol



Datanode always initiates connection!



Metadata Functionality

- Create Directory
- Delete Directory
- Write File
- Read File
- Delete File



Replication

- Reliability
- Read/Write Bandwidth
- Block Distribution



Replica Placement



Replica 1: same node as client (or random), rack A



Replica 2: a node in a different rack B



Replica 3: a node in same rack B

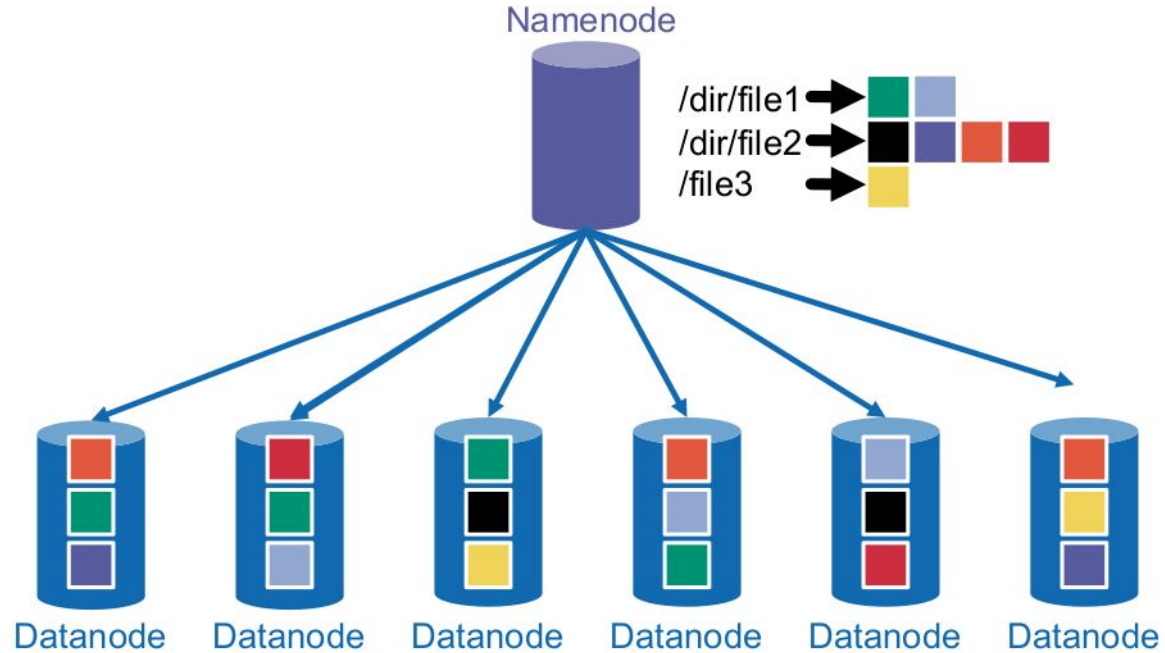


Replica 4 and beyond: random, but if possible:

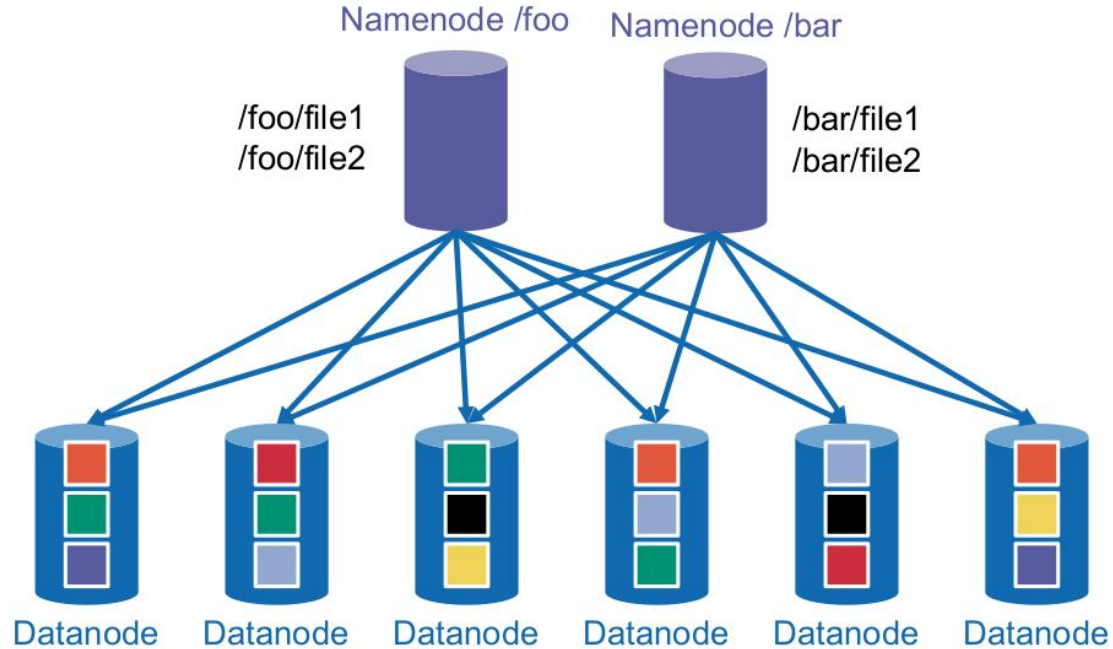
- at most one replica per node
- at most two replicas per rack



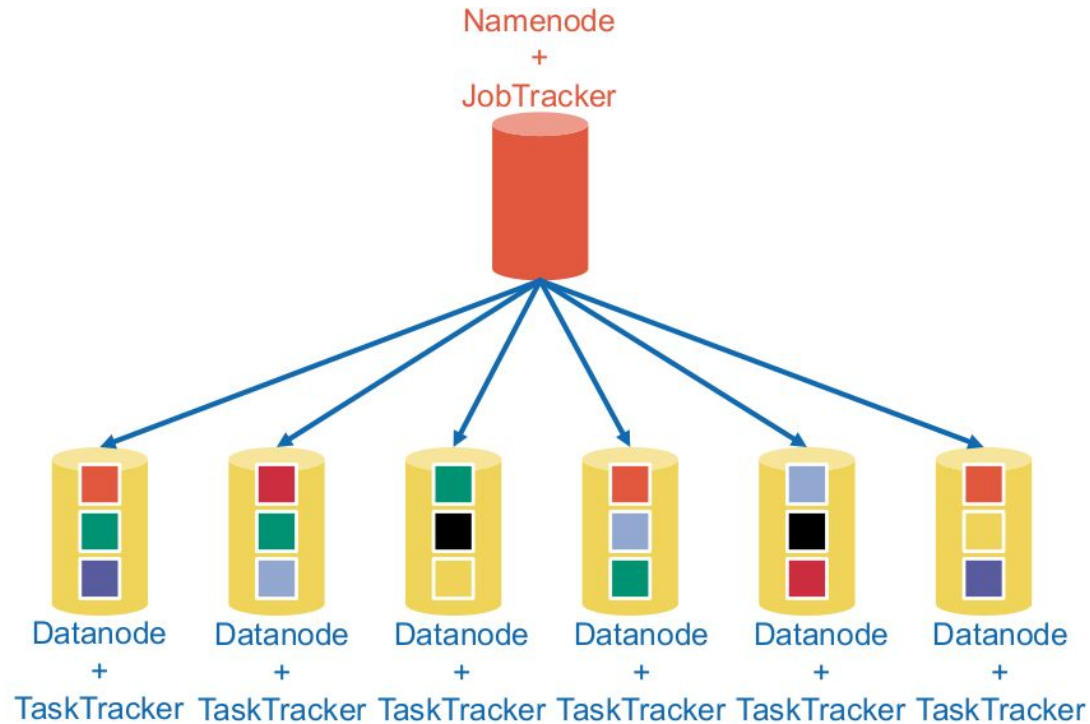
Single Point of Failure: NameNode



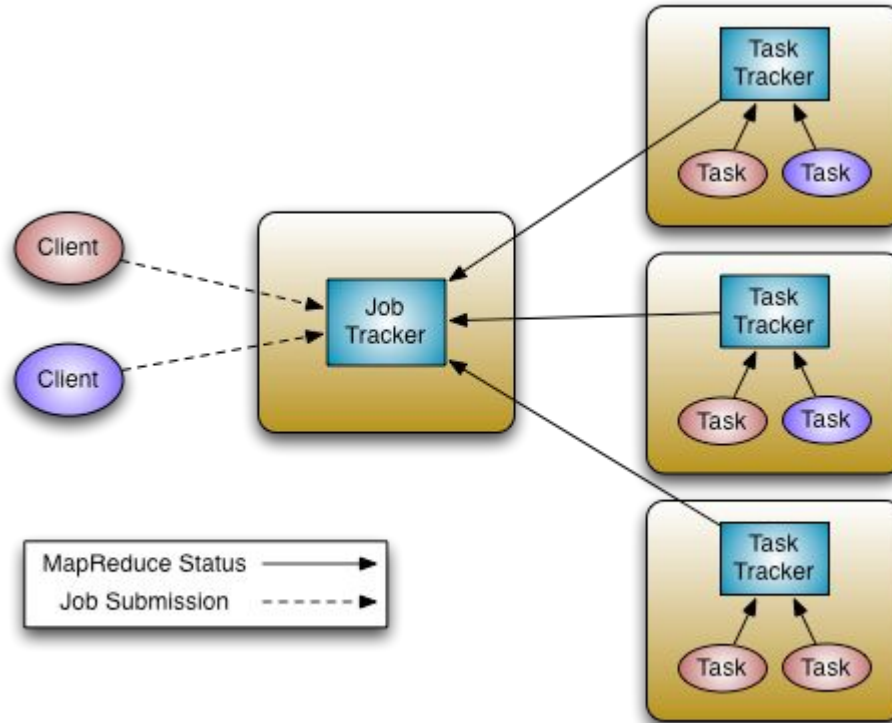
Secondary NameNode

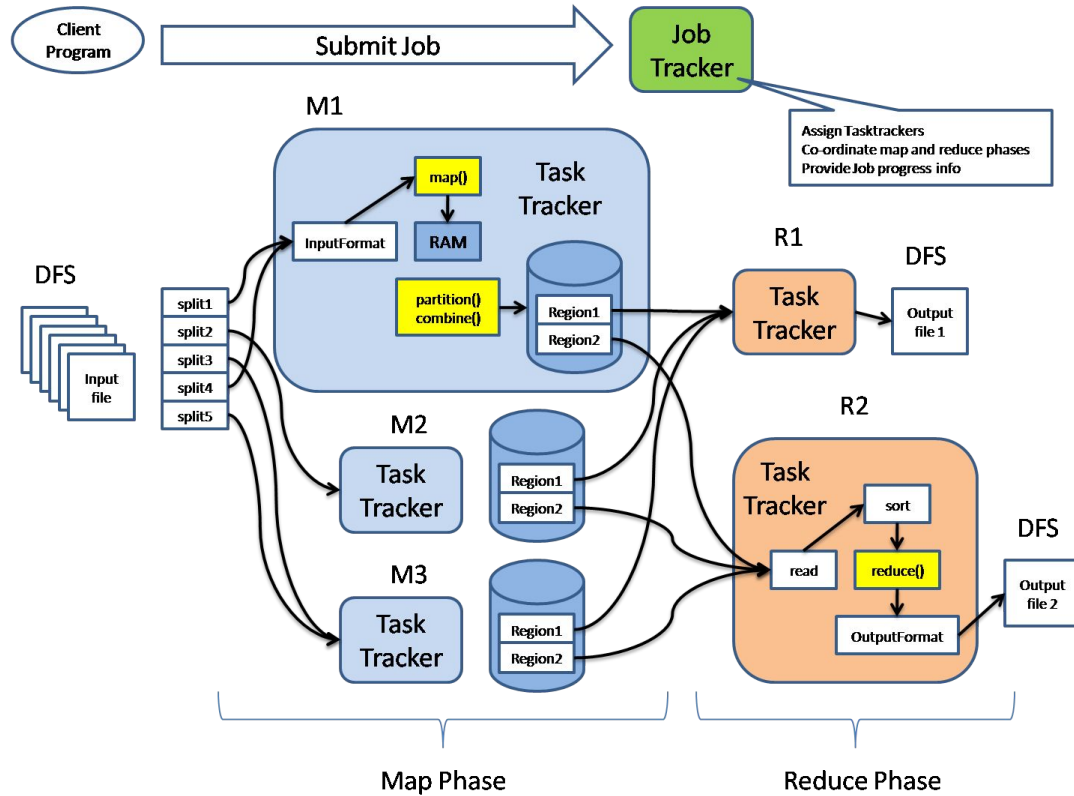


Hadoop MapReduce



Hadoop MapReduce Classic





Responsibilities of the MapReduce JobTracker



MapReduce Classic: Limitations

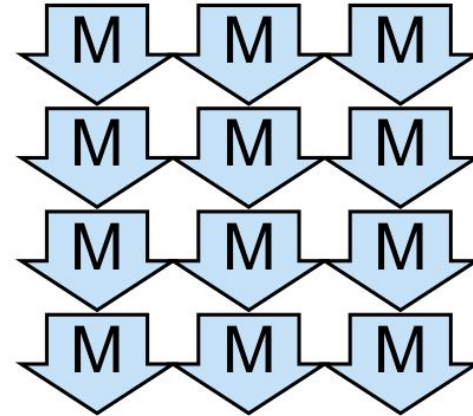
- Scalability
- Availability
- Resource Utilization
- Runs Only MapReduce Applications



Scalability



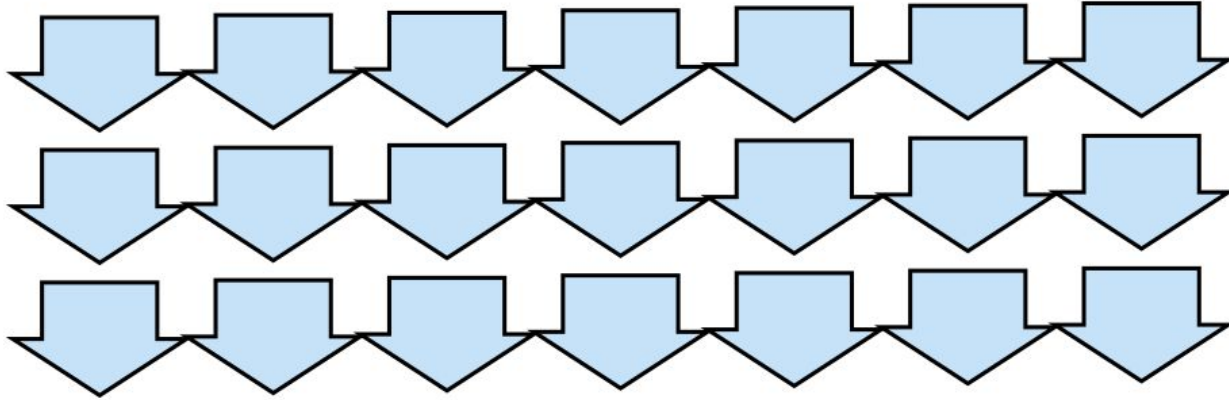
< 4,000 nodes



< 40,000 tasks



Utilization



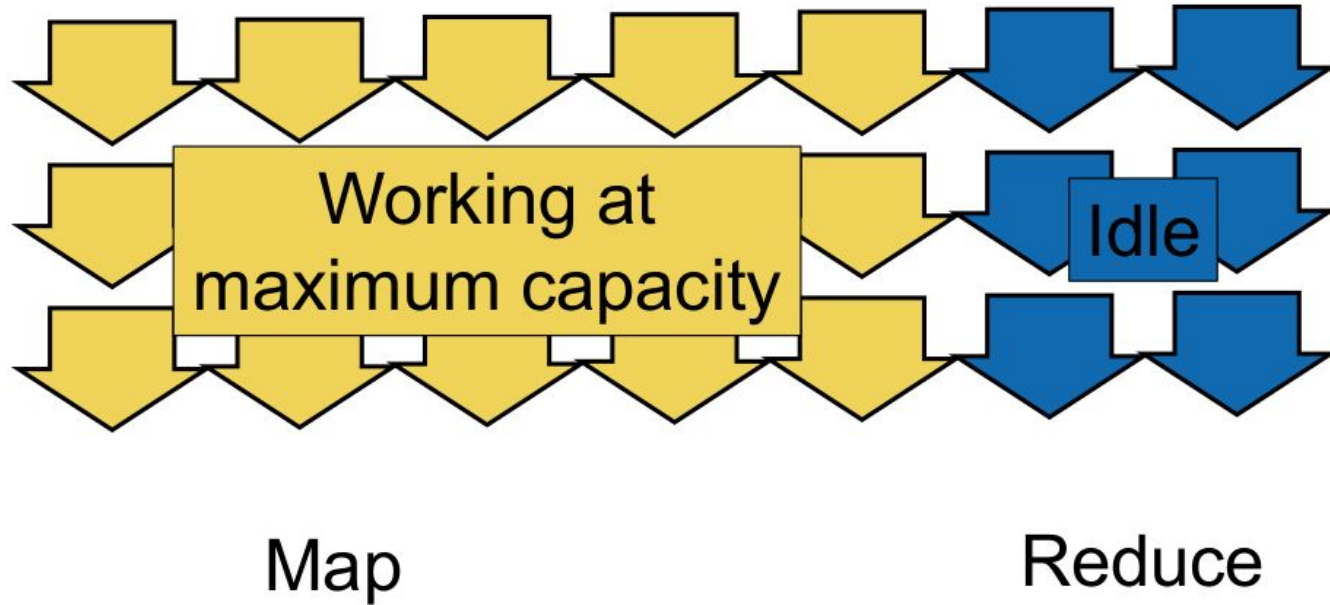
Static

(Decide on M/R at configuration time)

Fixed-size



Utilization

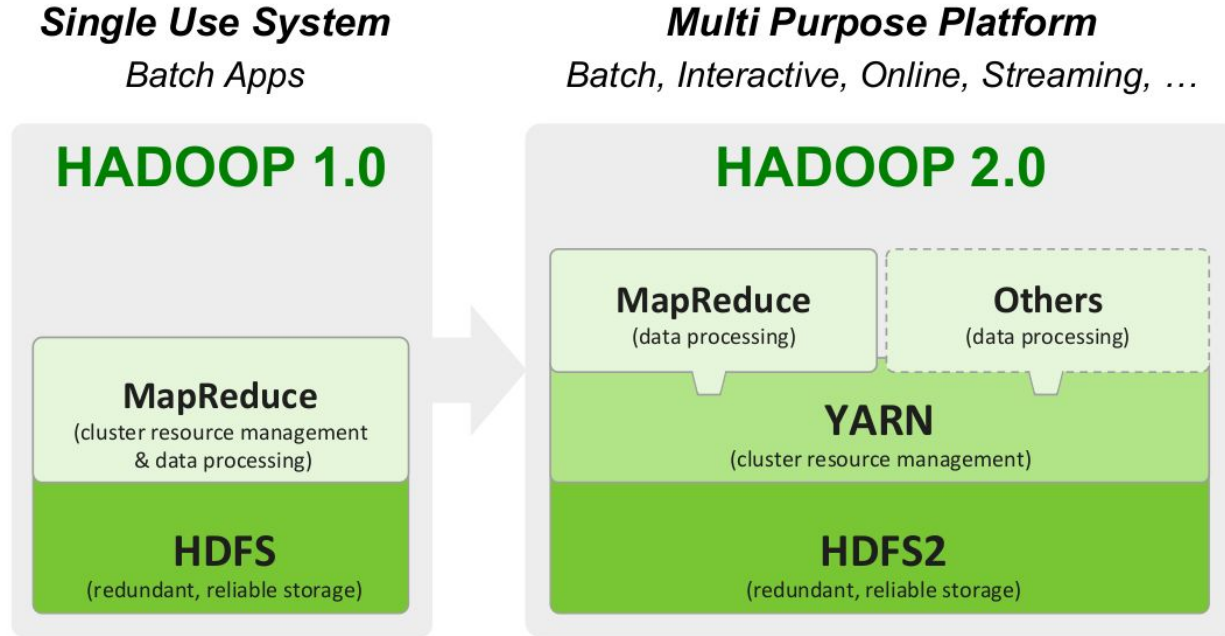


Apache Hadoop YARN



Data Engineering:
MapReduce, Hadoop and spark

The Fundamental Idea Behind YARN



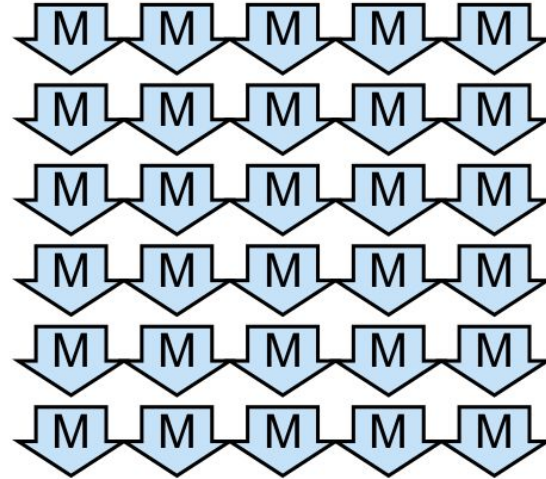
Applications Run Natively **IN** Hadoop



Scalability



10,000 nodes



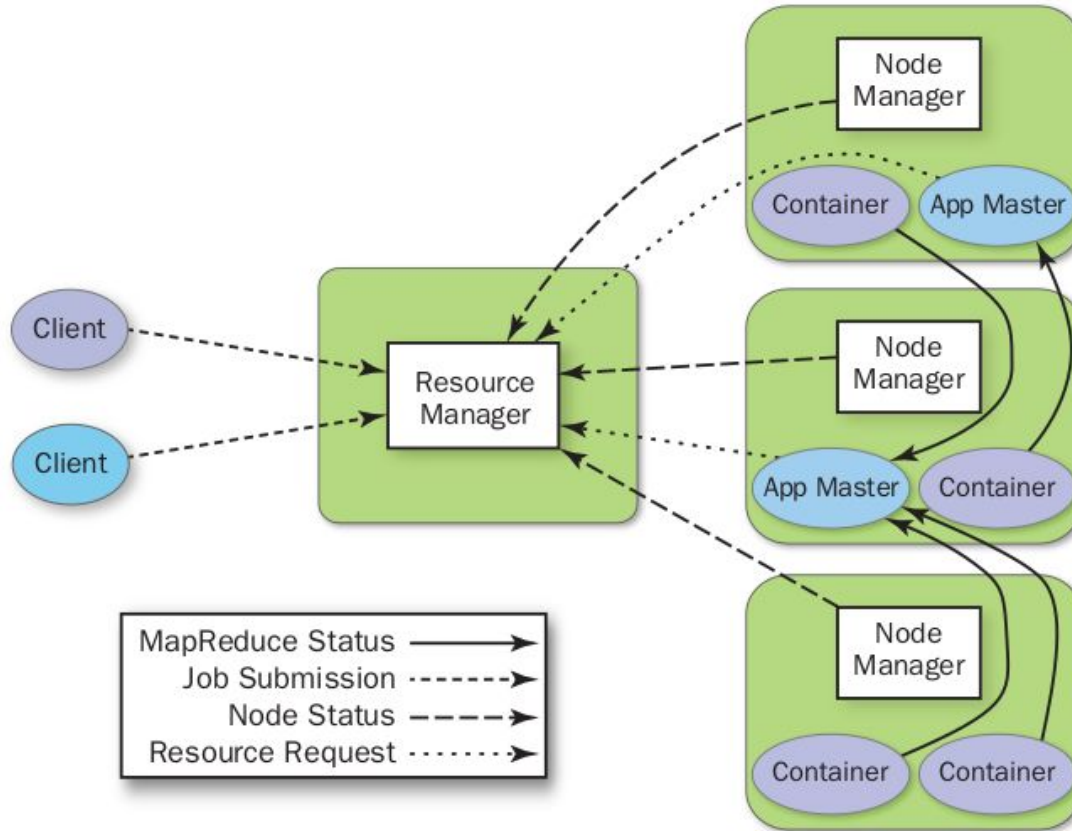
100,000 tasks

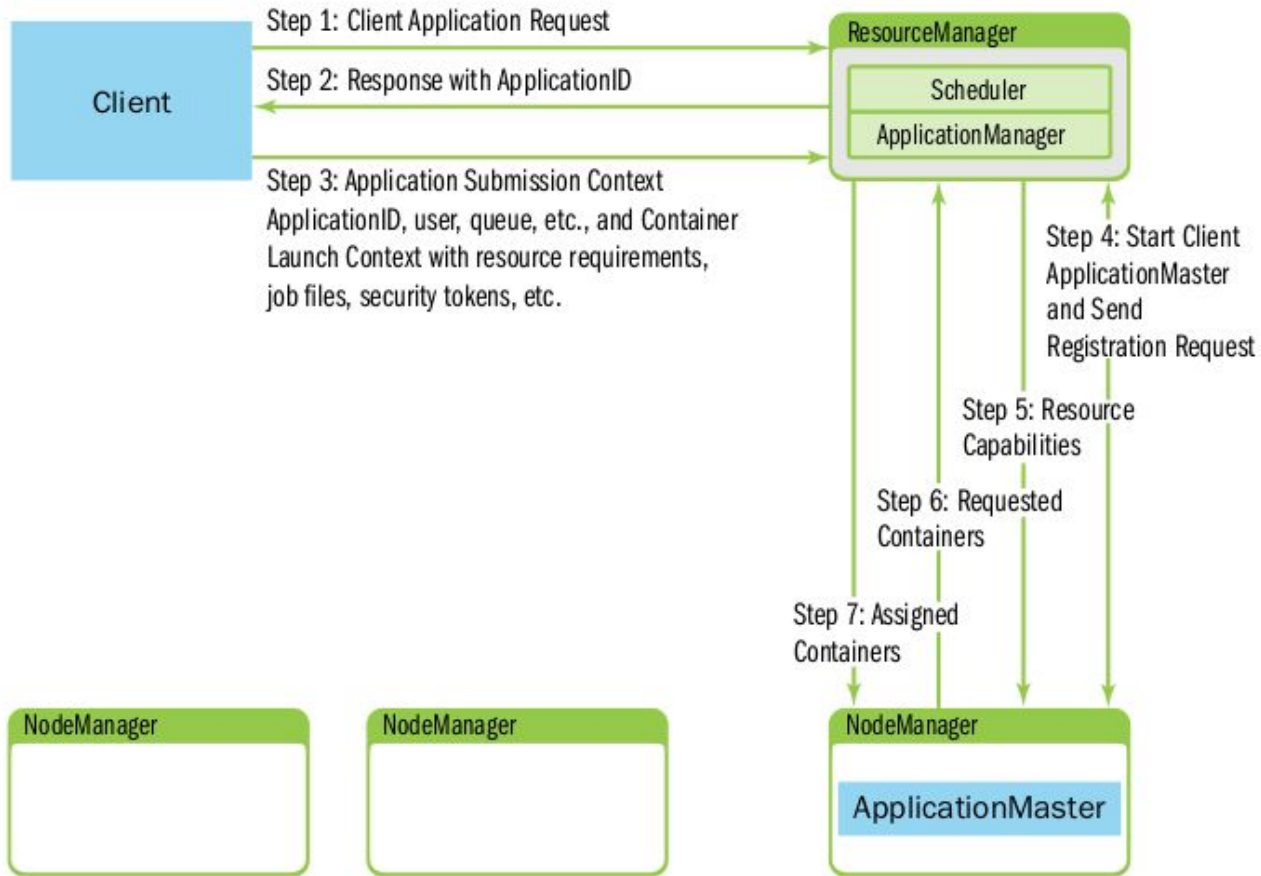


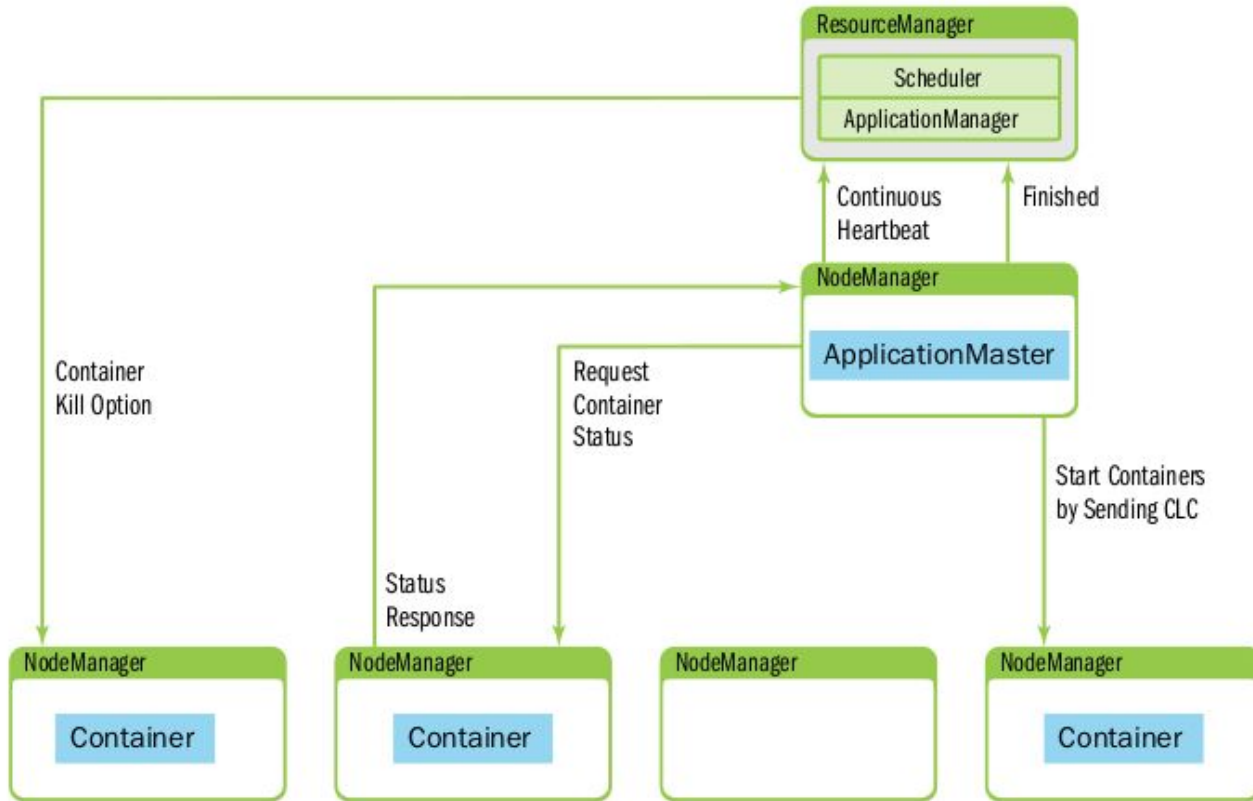
Hadoop v2 Architecture

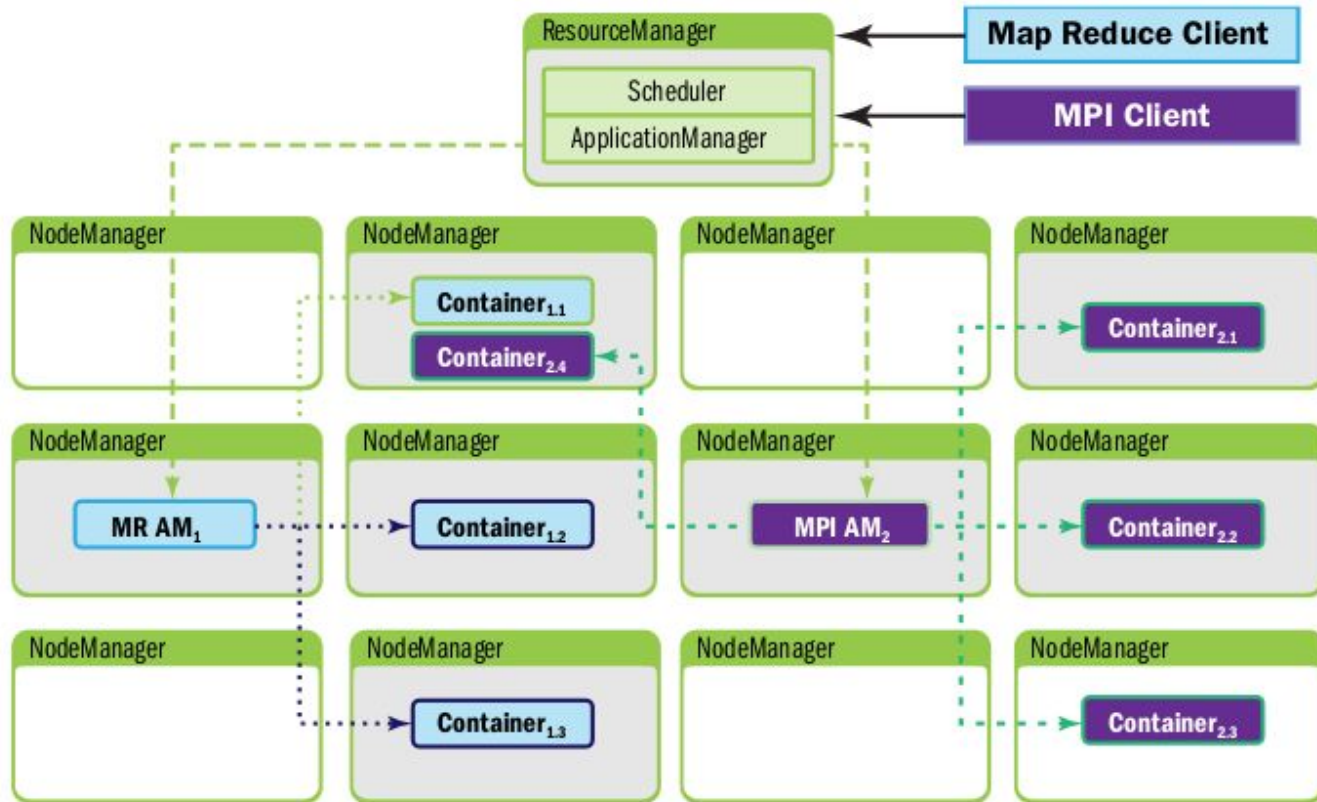
- Resource Manager
 - Manage and Allocates Cluster Resources
 - Application Scheduling
 - Application Manager
- Node Manager
 - Per-machine Agent
 - Manage Life-cycle of Container
 - Monitor Resources
- Application Master
 - Per-application
 - Manage Application Scheduling and Task Execution











Scheduling Algorithms

- FIFO Scheduler
- Capacity Scheduler
- Fair Scheduler

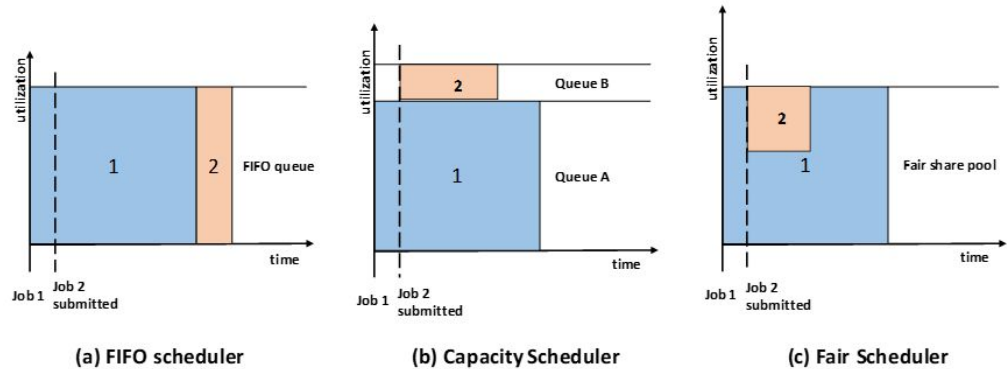


Figure 1: YARN Schedulers' cluster utilization vs. time





Berkeley design of MapReduce programming.

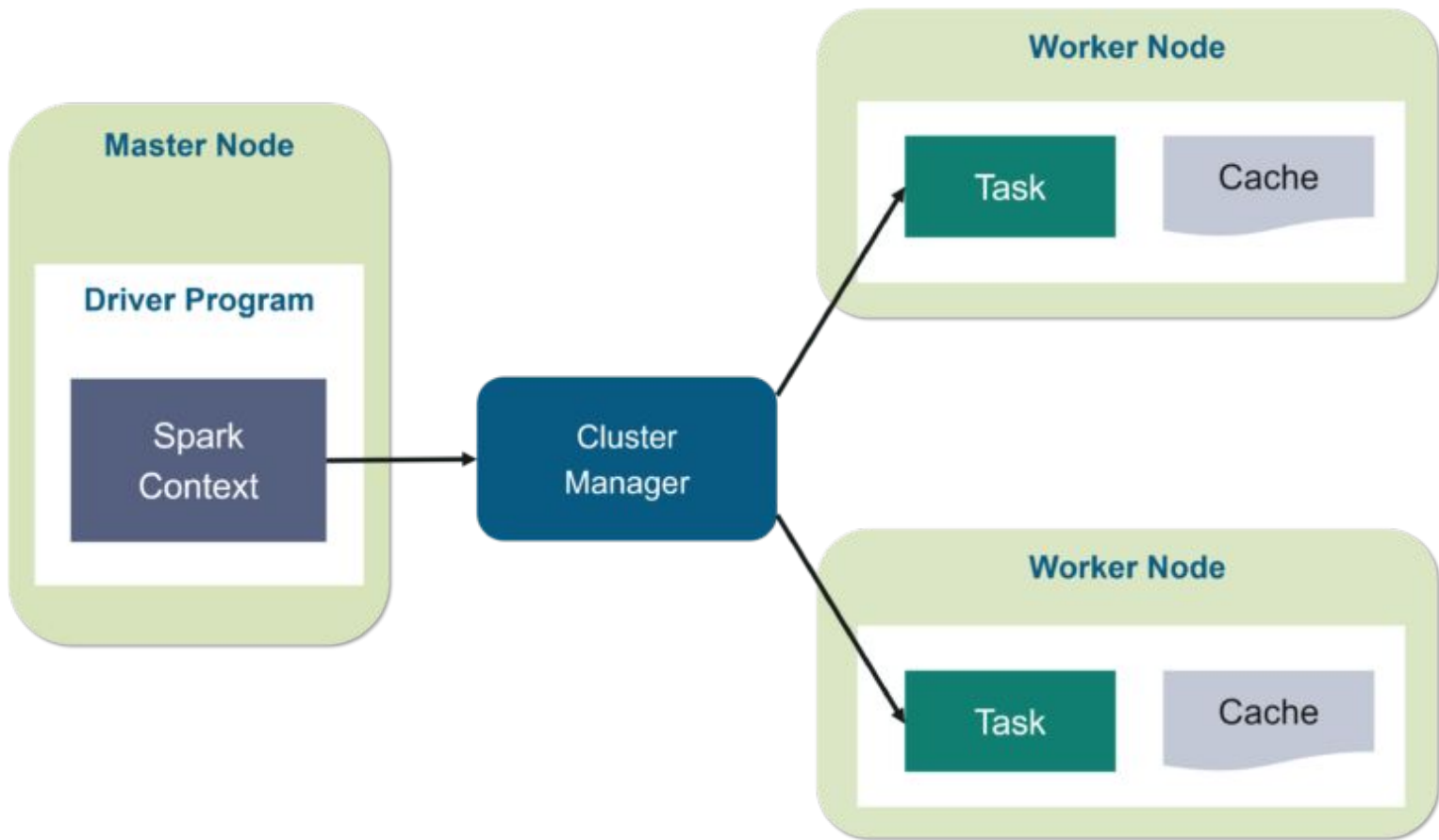


Data Engineering:
MapReduce, Hadoop and spark





Data Engineering:
MapReduce, Hadoop and spark



Spark Architecture Overview

- Resilient Distributed Dataset (RDD)
- Directed Acyclic Graph (DAG)

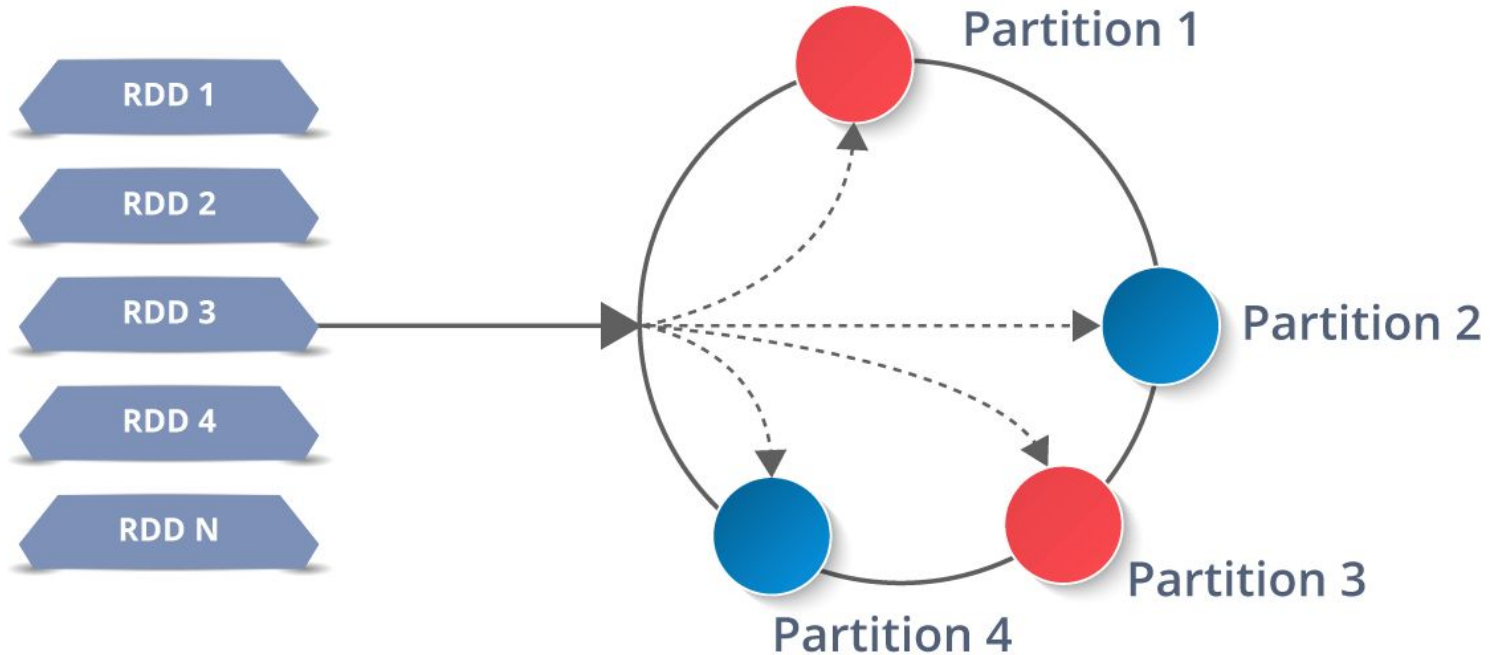


Resilient Distributed Datasets (RDDs):

Read-only, partitioned collection
of records.



Resilient Distributed Dataset (RDD)

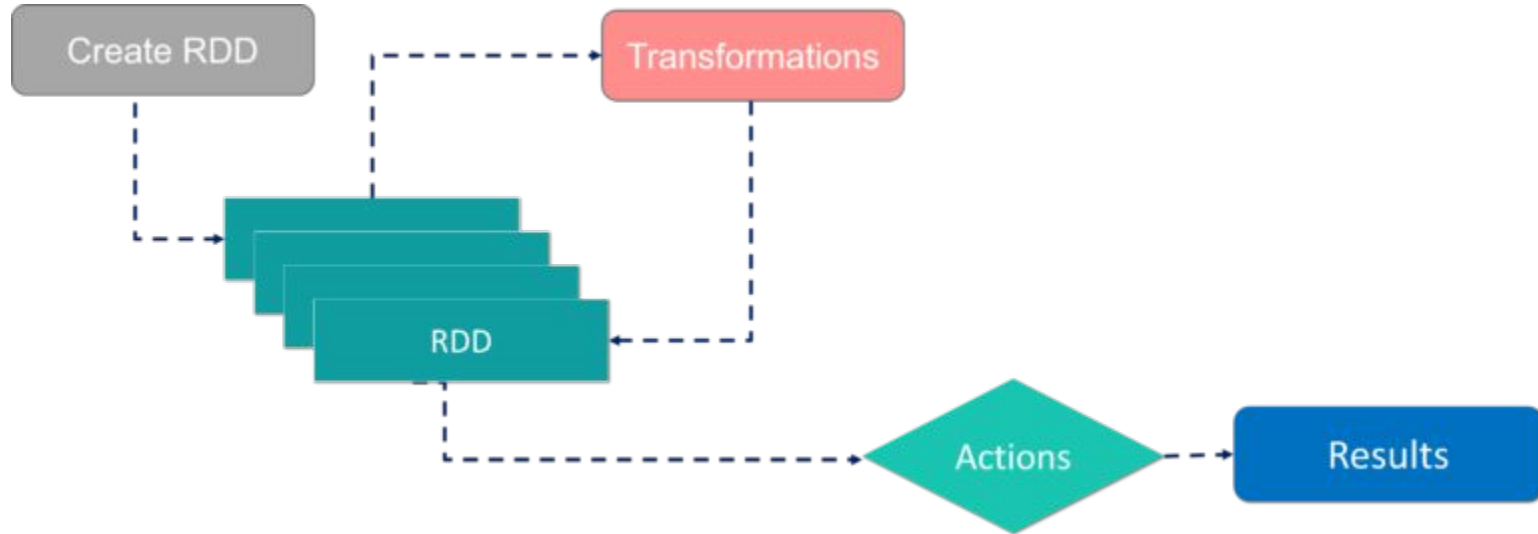


Operators

- Transformations
- Actions



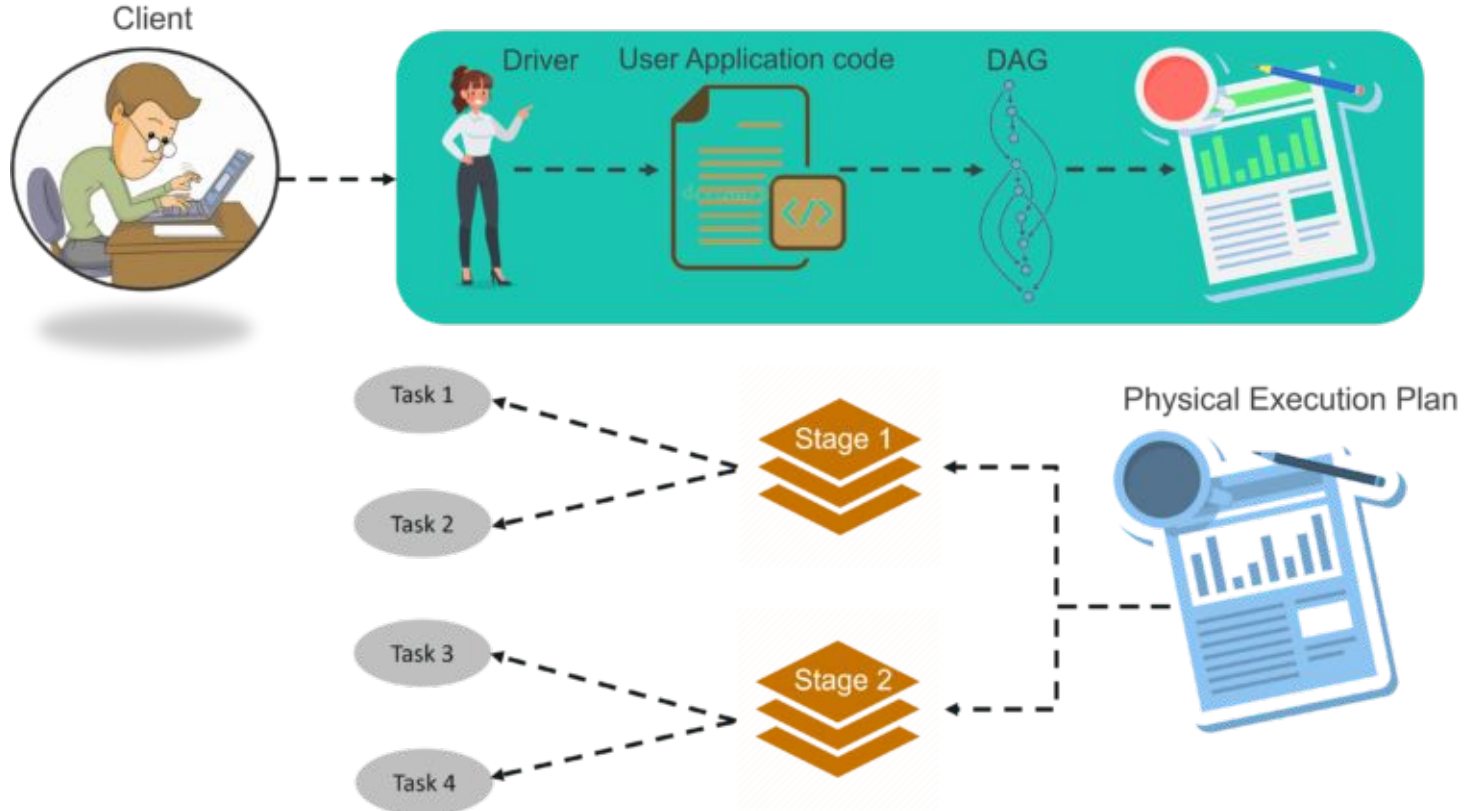
Resilient Distributed Dataset (RDD)



Directed Acyclic Graph (DAG)

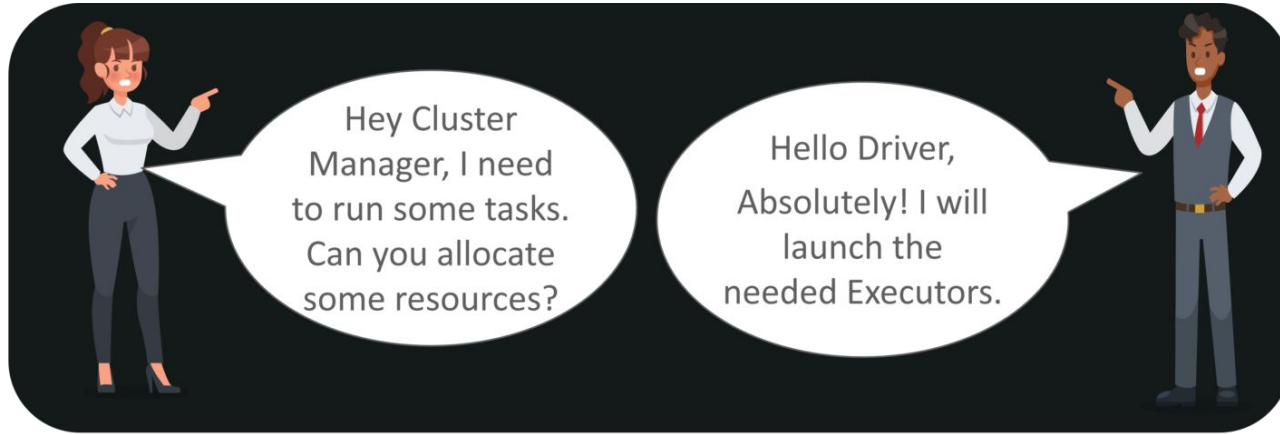


Data Engineering:
MapReduce, Hadoop and spark

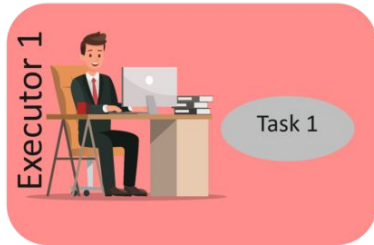


Data Engineering:
MapReduce, Hadoop and spark

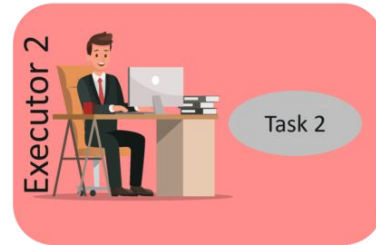




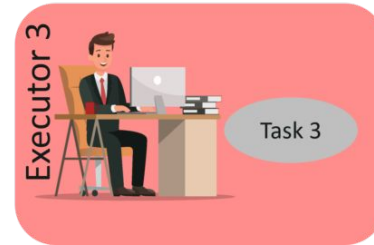
Worker 1



Worker 2



Worker 3



Worker 4





References

- [Apache Spark architecture](#)
- [Big Data for Engineers 2018-ETH course](#)
- MapReduce: Simplified Data Processing on Large Clusters-Jeffrey Dean and Sanjay Ghemawat
- Fundamentals of Database Systems-Ramez Elmasri and Shamkant B. Navathe
- Hadoop Platform and Application Framework-Coursera
- Real-Time Big Data Analytics: Emerging Architecture-Mike Barlow



References

- Apache Hadoop YARN-Arun C. Murthy and etc.
- Apache Hadoop YARN: Yet Another Resource Negotiator-Vinod Kumar Vavilapalli and etc.
- Apache Spark-CS240A, Winter 2016. T Yang



