

# Hadoop

- Open source software framework designed for storage and processing of large scale data on clusters of commodity hardware.
- Created by Doug Cutting and Mike Carafella in 2005.
- Based on work done by Google in the early 2000s
  - “The Google File System” in 2003
  - “MapReduce: Simplified Data Processing on Large Clusters” in 2004
- Cutting named the program after his son’s toy elephant.

# Hadoop's Developers



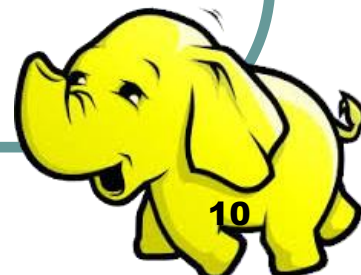
Doug Cutting



**2005:** Doug Cutting and Michael J. Cafarella developed Hadoop to support distribution for the Nutch search engine project.

The project was funded by Yahoo.

**2006:** Yahoo gave the project to Apache Software Foundation.



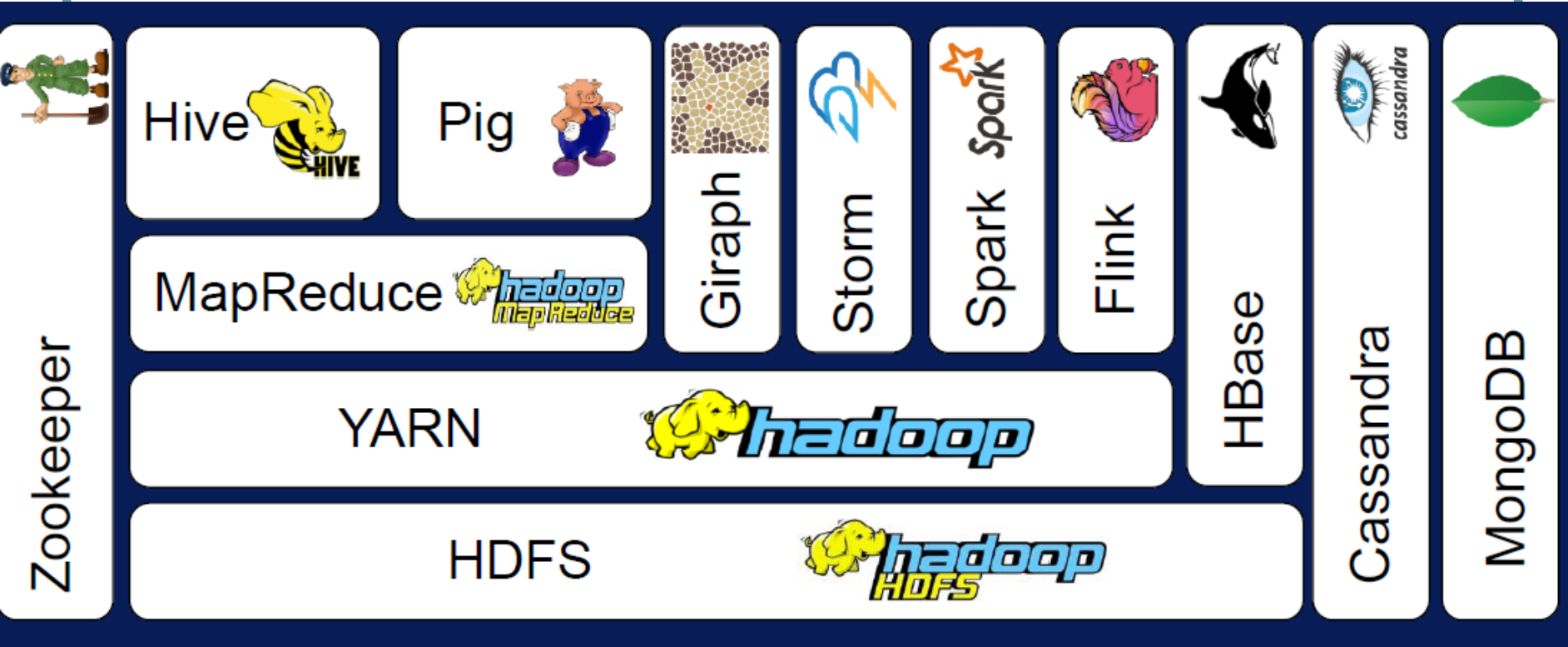
# What is Hadoop?

- At Google MapReduce operation are run on a GFS
- GFS is not open source.
- Doug Cutting and Yahoo! reverse engineered the GFS and called it Hadoop Distributed File System (HDFS).
- The software framework that supports HDFS, MapReduce and other related entities is called the project Hadoop or simply Hadoop.
- This is open source and distributed by Apache.

# The Hadoop Ecosystem

- Four modules:
  - **Hadoop Common**: a set of shared programming libraries used by the other modules
  - **Hadoop Distributed File System (HDFS)**: a Java-based file system to store data across multiple machines
  - **MapReduce framework**: a programming model to process large sets of data in parallel
  - **YARN (Yet Another Resource Negotiator)**: handles the management and scheduling of resource requests in a distributed environment

# The Hadoop Ecosystem



# HDFS

- HDFS is a file system written in Java based on the Google's GFS
- Responsible for storing data on the cluster
- Provides redundant storage for massive amounts of data
- Data files are split into blocks and distributed across the nodes in the cluster
- Each block is replicated multiple times
- HDFS cluster is composed of a NameNode and various DataNodes

# HDFS

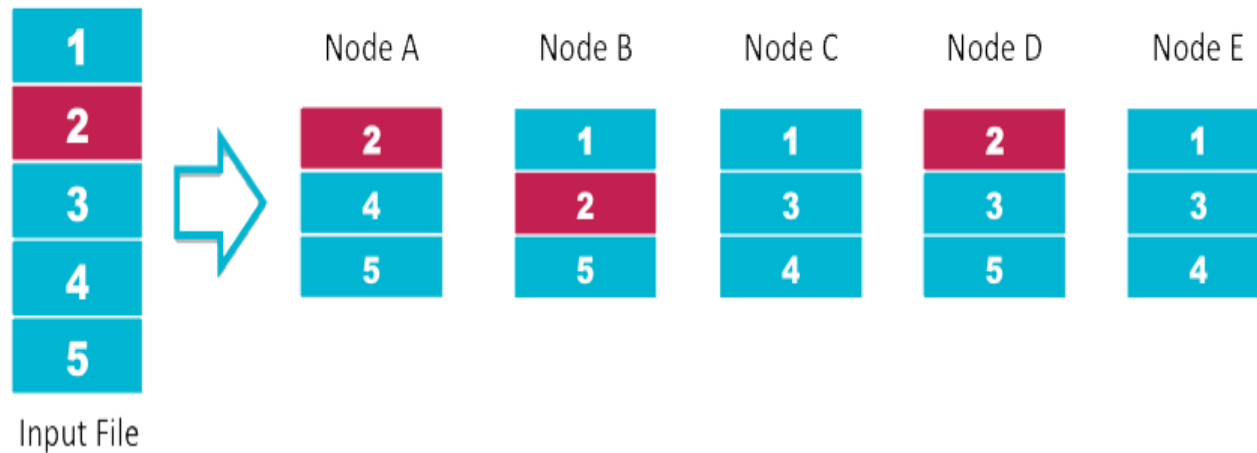
- **NameNode**
  - a server which holds all the metadata regarding the stored files
  - manages incoming file system operations
  - maps data blocks (parts of files) to DataNodes
- **DataNode**
  - handles file read and write requests
  - create, delete and replicate data blocks amongst their disk drives
  - continuously loop, asking the NameNode for instructions.
- **Note:** size of 1 data block is typically 128 megabytes



# HDFS: Data Replication

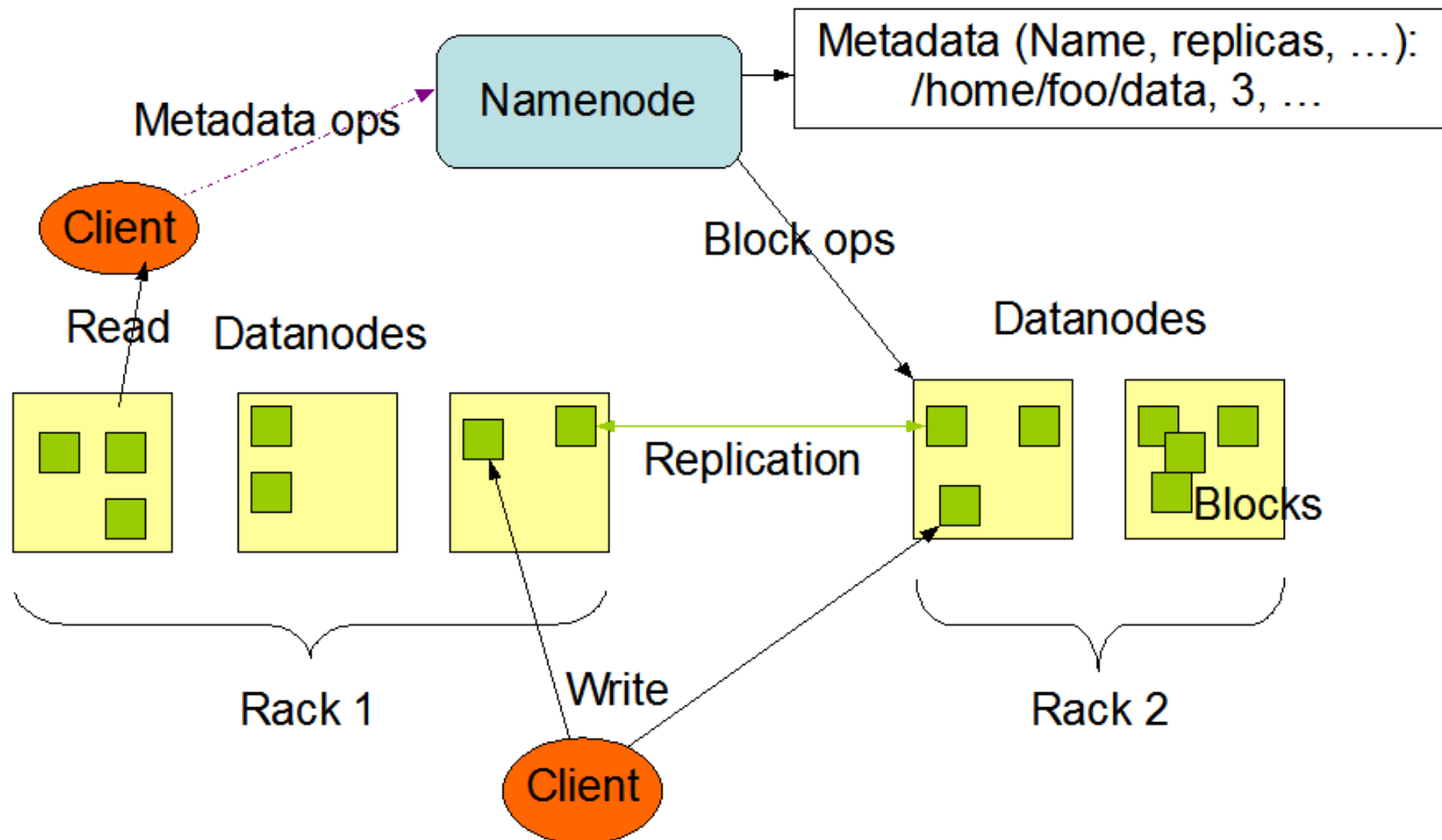
- Default replication is 3-fold

HDFS Data Distribution



# HDFS Architecture

HDFS Architecture



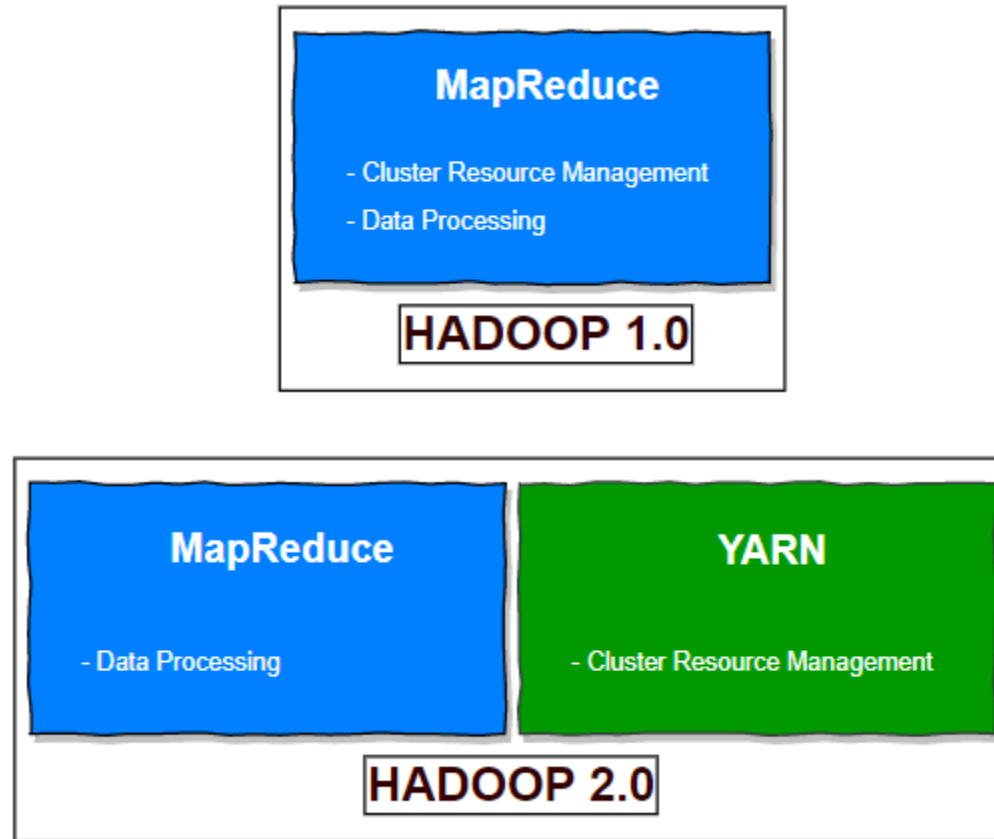
# YARN

- Yet Another Resource Negotiator
- YARN Application Resource Negotiator(Recursive Acronym)
- Remedies the scalability shortcomings of “classic” MapReduce
- Is more of a general purpose framework of which classic mapreduce is one application.

# Hadoop YARN

- A framework for job scheduling and cluster resource management.
- In 2012, YARN became a sub-project of the larger Apache Hadoop project.
- Sometimes called MapReduce 2.0, YARN is a software rewrite that decouples MapReduce's resource management and scheduling capabilities from the data processing component, enabling Hadoop to support more varied processing approaches and a broader array of applications.

# Yet Another Resource Negotiator (YARN)



# Yet Another Resource Negotiator (YARN)

- Yet Another Resource Negotiator (YARN) distributes a MapReduce program across different nodes and takes care of coordination
- Three important services
  - **ResourceManager**: a global YARN service that receives and runs applications (e.g., a MapReduce job) on the cluster
  - **JobHistoryServer**: keeps a log of all finished jobs
  - **NodeManager**: responsible to oversee resource consumption on a node

# Hadoop Daemons

- Hadoop consist of five daemons.
  - **NameNode**
  - **DataNode**
  - **Secondary nameNode**
  - **Resource Manager**
  - **Node Manager**
- “Running Hadoop” means running a set of daemons, or resident programs, on the different servers in your network.
- These daemons have specific roles; some exist only on one server, some exist across multiple servers.

# Hadoop Configuration Modes

- **Local (standalone) mode**

- The standalone mode is the default mode for Hadoop.
- Hadoop chooses to be conservative and assumes a minimal configuration. All XML (Configuration) files are empty under this default mode.
- With empty configuration files, Hadoop will run completely on the local machine.
- Because there's no need to communicate with other nodes, the standalone mode doesn't use HDFS, nor will it launch any of the Hadoop daemons.
- Its primary use is for developing and debugging the application logic of a Map-Reduce program without the additional complexity of interacting with the daemons.



# Hadoop Configuration Modes

- **Pseudo-distributed mode**

- The pseudo-distributed mode is running Hadoop in a “cluster of one” with all daemons running on a single machine.
- This mode complements the standalone mode for debugging your code, allowing you to examine memory usage, HDFS input/output issues, and other daemon interactions.
- Need Configuration on XML Files.

# Hadoop Configuration Modes

- **Fully distributed mode**


- Benefits of distributed storage and distributed computation
- **master**—The server that hosts the NameNode and Resorcemanager daemons
- **backup**—The server that hosts the Secondary NameNode daemon
- **slaves**—The servers that host both DataNode and Nodemanager daemons

# Uses for Hadoop

- Data-intensive text processing
- Assembly of large genomes
- Graph mining
- Machine learning and data mining
- Large scale social network analysis

# Who Uses Hadoop?





# Thank You !