COSC 3337 : Data Science I

N. Rizk

College of Natural and Applied Sciences

Department of Computer Science

University of Houston

Hadoop

A distributed framework for <u>Big Data</u>

Designed to answer the question:

"How to process big data with reasonable cost and time?"

Problem

- How do you scale up applications?
  - Run jobs processing 100's of terabytes of data
  - Takes 11 days to read on 1 computer
- Need lots of cheap computers
  - Fixes speed problem (15 minutes on 1000 computers), but…
  - Reliability problems
    - In large clusters, computers fail every day
    - Cluster size is not fixed
- Need common infrastructure
  - Must be efficient and reliable

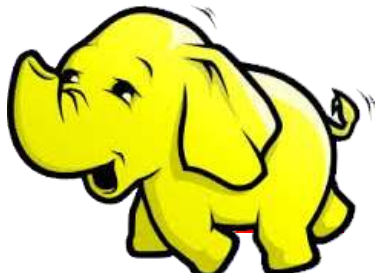
Solution

- Open Source Apache Project
- Hadoop Core includes:
  - Distributed File System - distributes data
  - Map/Reduce - distributes application
- Written in Java
- Runs on
  - Linux, Mac OS/X, Windows, and Solaris
  - Commodity hardware

The Apache Software Foundation /əˈpætʃi/ (**ASF**) is an American nonprofit corporation (classified as a 501(c)(3) organization in the United States) to support Apache software projects, including the Apache HTTP Server. The ASF was formed from the Apache Group and incorporated on March 25, 1999.

What is Hadoop?

- **Hadoop:**
  - an open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license.
- **Goals / Requirements:**
  - Abstract and facilitate the storage and processing of large and/or rapidly growing data sets
    - Structured and non-structured data
    - Simple programming models
  - High scalability and availability
  - Use commodity (cheap!) hardware with little redundancy
  - Fault-tolerance
  - Move computation rather than data



What is Hadoop?

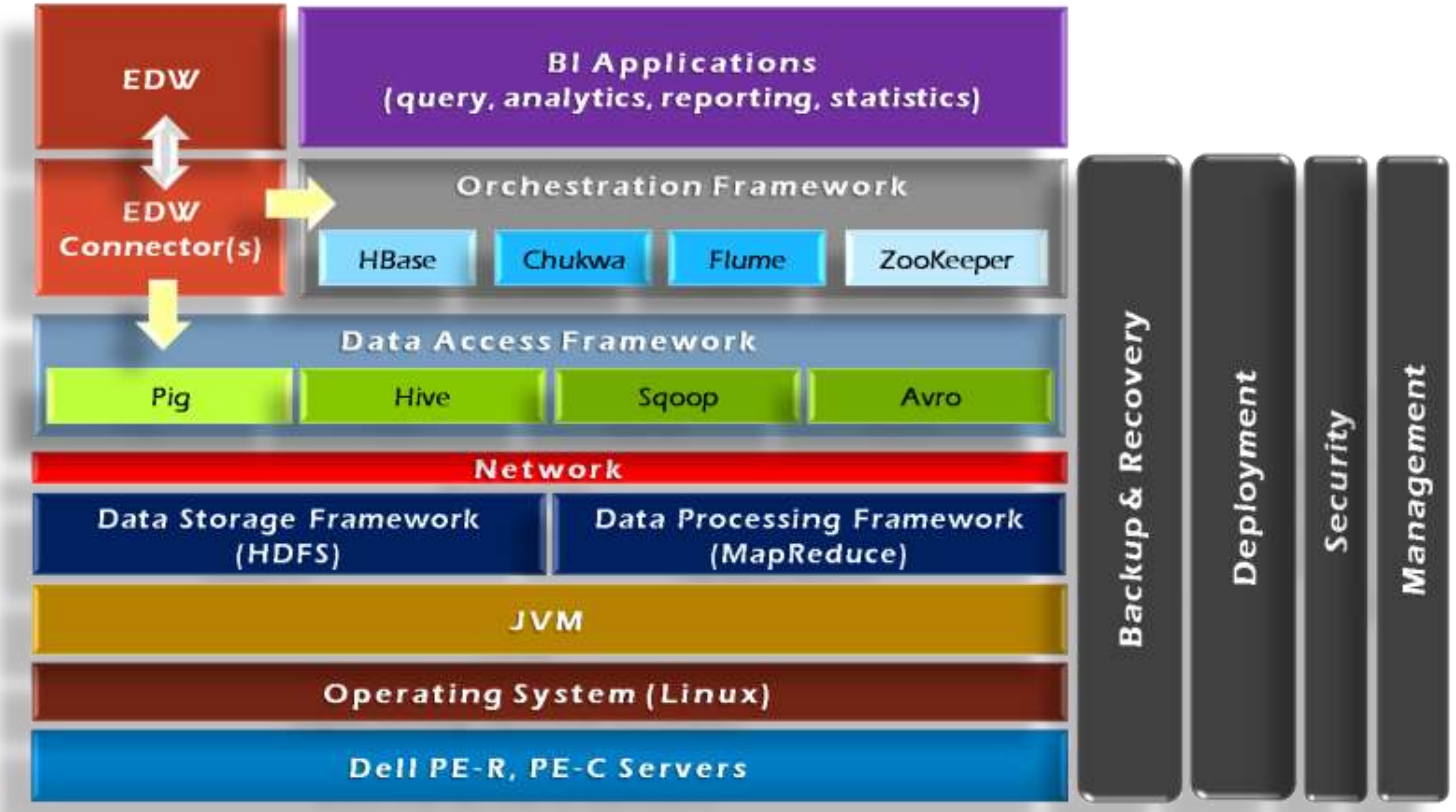
- Apache top level project, open-source implementation of frameworks for reliable, scalable, distributed computing and data storage.

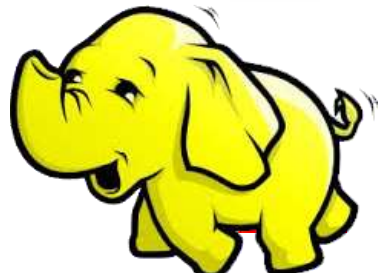
- It is a flexible and highly-available architecture for large scale computation and data processing on a network of commodity hardware.

Hadoop Framework Tools

- **2008 - Hadoop Wins Terabyte Sort Benchmark (**sorted 1 terabyte of data in 209 seconds, compared to previous record of 297 seconds)
- 2009 - Avro and Chukwa became new members of Hadoop Framework family
- 2010 - Hadoop's Hbase, Hive and Pig subprojects completed, adding more computational power to Hadoop framework
- **2011 - ZooKeeper Completed**
- **2013 - Hadoop 1.1.2 and Hadoop 2.0.3 alpha.**

- Ambari, Cassandra, Mahout have been added





Hadoop's Developers

Doug Cutting

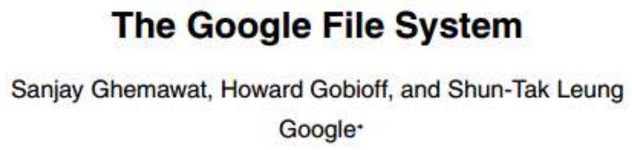
2005: Doug Cutting and Michael J. Cafarella developed Hadoop to support distribution for the Nutch search engine project.

The project was funded by Yahoo.

2006: Yahoo gave the project to Apache Software Foundation.

Google Origins

2003

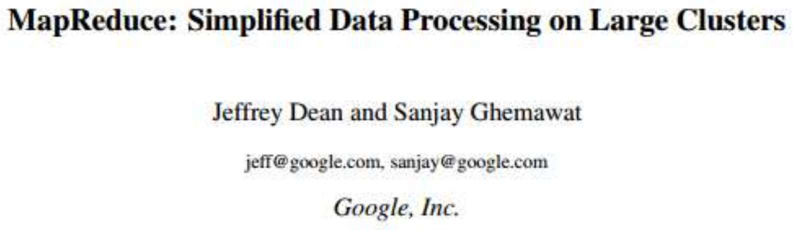


The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google*

2004



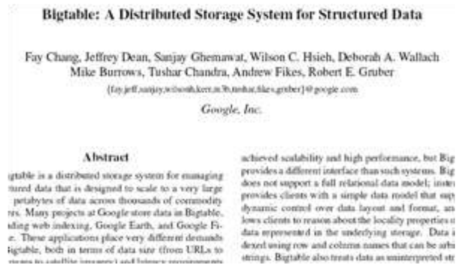
MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

2006



Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber

{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com

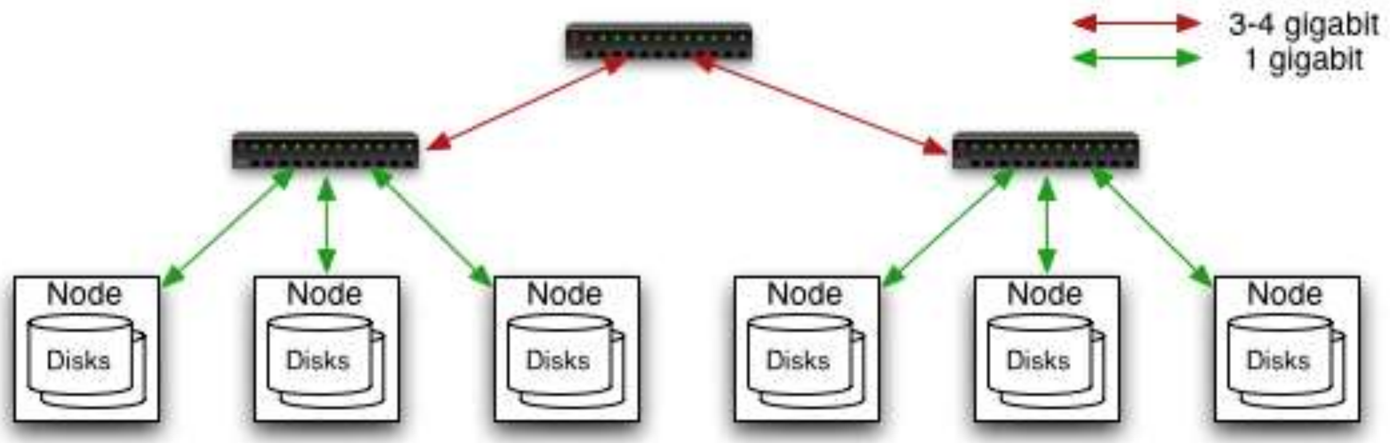
Google, Inc.

Abstract

gtable is a distributed storage system for managing
tured data that is designed to scale to a very large
petabytes of data across thousands of commodity
rs. Many projects at Google store data in Bigtable,
ding web indexing, Google Earth, and Google Fi-
e. These applications place very different demands
Bigtable, both in terms of data size (from URLs to

achieved scalability and high performance, but Big
provides a different interface than such systems. Big
does not support a full relational data model; inste
provides clients with a simple data model that sup
dynamic control over data layout and format, an
lows clients to reason about the locality properties o
data represented in the underlying storage. Data i
dexed using row and column names that can be arbi
strings. Bigtable also treats data as uninterpreted str

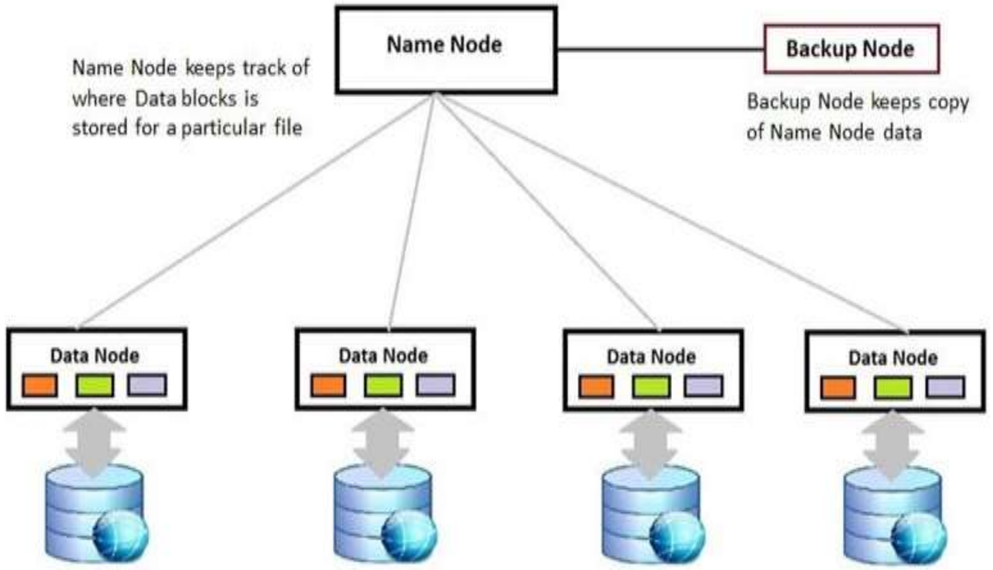
Commodity Hardware Cluster



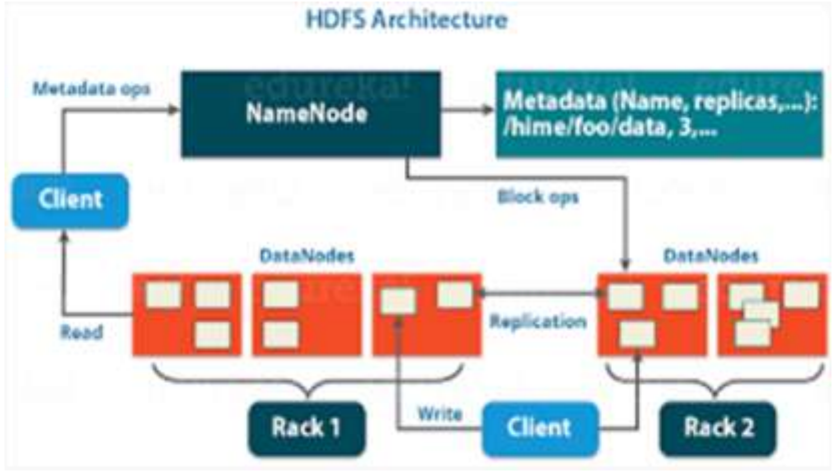
- Typically in 2 level architecture
  - Nodes are commodity PCs
  - 40 nodes/rack
  - Uplink from rack is 8 gigabit
  - Rack-internal is 1 gigabit

Distributed File System

- Single namespace for entire cluster
  - Managed by a single *namenode.*
  - Files are single-writer and append-only.
  - Optimized for streaming reads of large files.
- Files are broken into large blocks.
  - Typically 128 MB
  - Replicated to several *datanodes*, for reliability
- Client talks to both namenode and datanod
  - Data is not sent through the namenode.
  - Throughput of file system scales nearly linearly with the number of nodes.
- Access from Java, C, or command line.



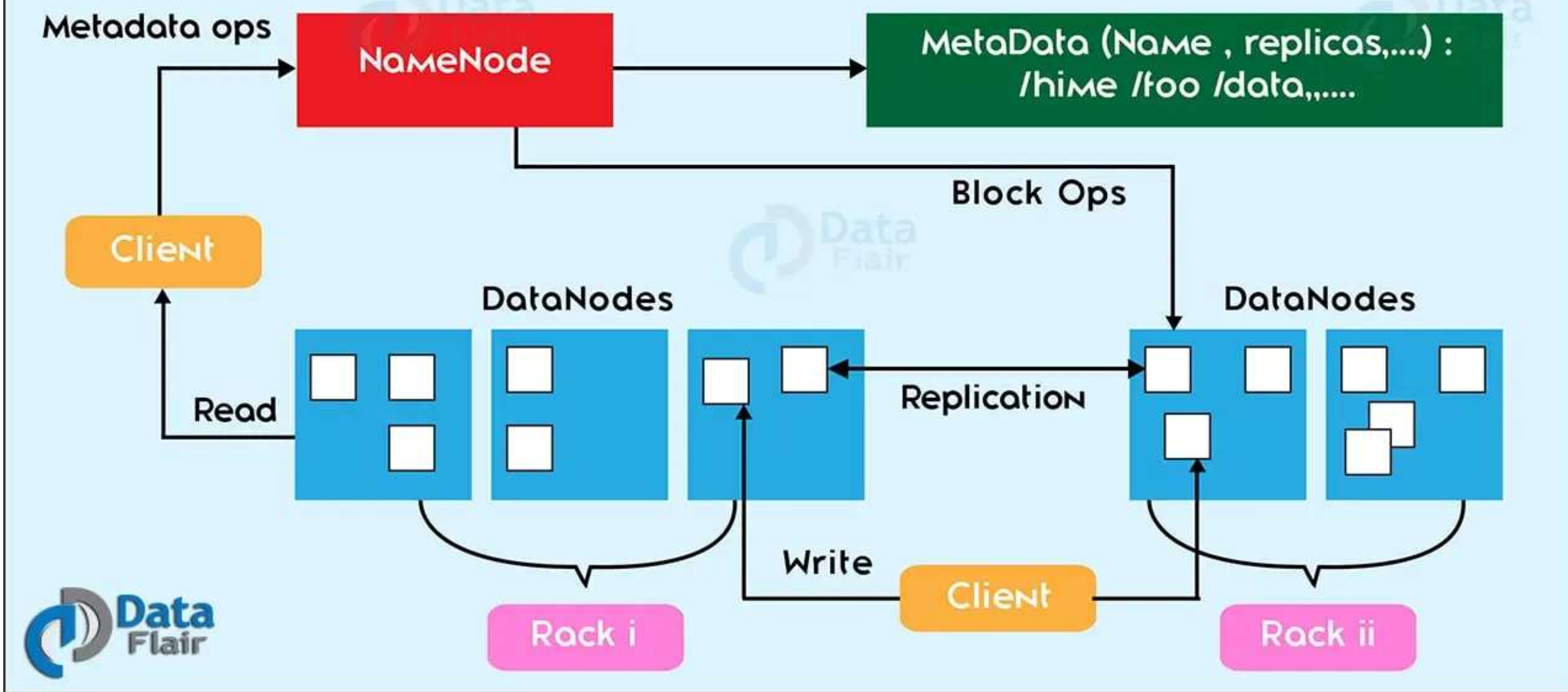
Name Node
Backup Node
Name Node keeps track of where Data blocks is stored for a particular file
Backup Node keeps copy of Name Node data
Data Node
Data Node
Data Node
Data Node



Block Placement

- Default is 3 replicas, but settable
- Blocks are placed (writes are pipelined):
  - On same node
  - On different rack
  - On the other rack
- Clients read from closest replica
- If the replication for a block drops below target, it is automatically re-replicated.

HDFS Architecture



Data Correctness

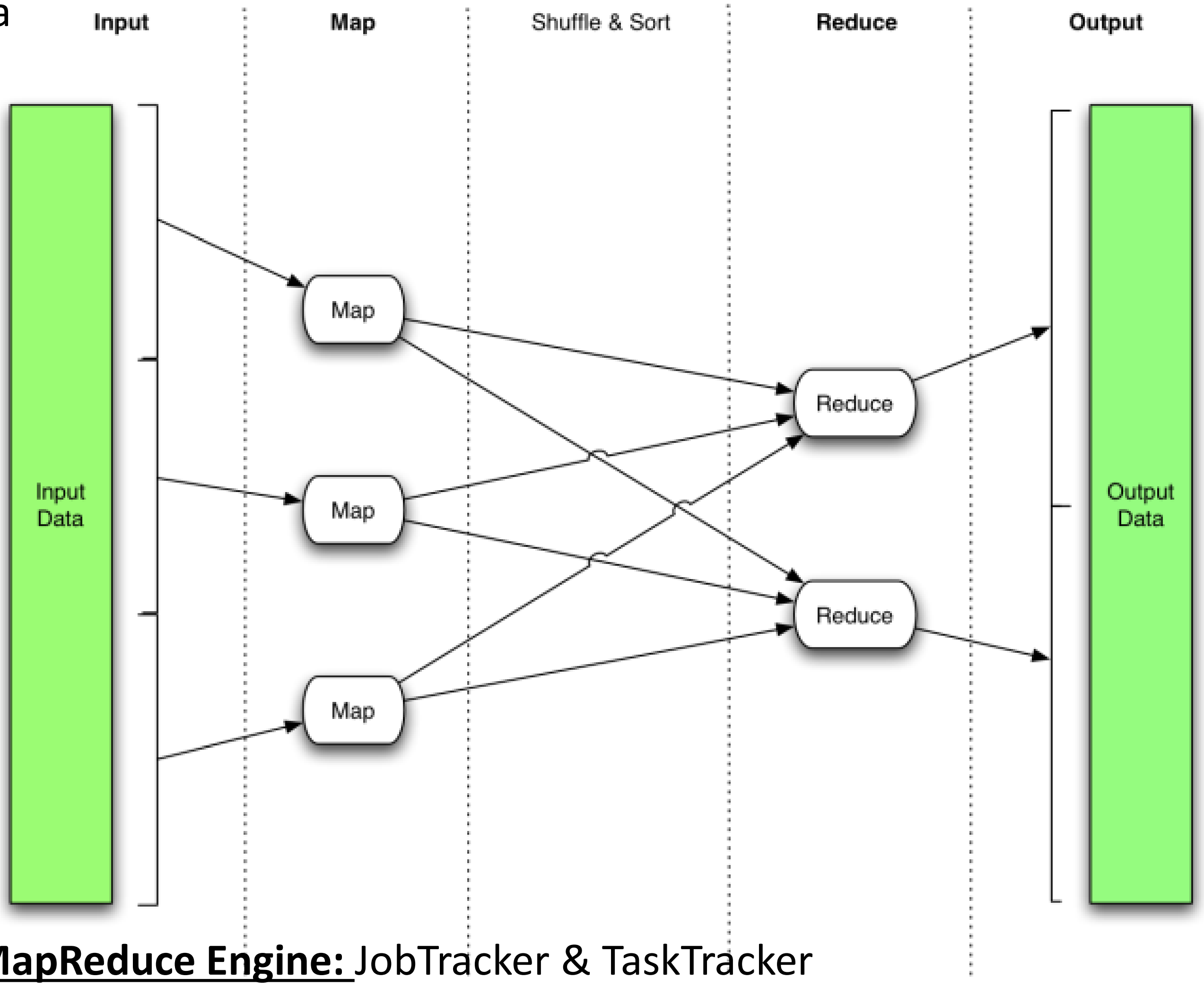
- Data is checked with CRC32
- File Creation
    - Client computes checksum per 512 byte
    - DataNode stores the checksum
- File access
    - Client retrieves the data and checksum from DataNode
    - If Validation fails, Client tries other replicas
- Periodic Validation

Map/Reduce

- Map/Reduce is a programming model for efficient distributed computing
- It works like a Unix pipeline:
  - cat input | grep | sort | uniq -c | cat > output
  - **Input** | **Map** | Shuffle & Sort | **Reduce** | **Output**
- Efficiency from
  - Streaming through data, reducing seeks
  - Pipelining
- A good fit for a lot of applications
  - Log processing
  - Web index building

Map/Reduce Dataflow

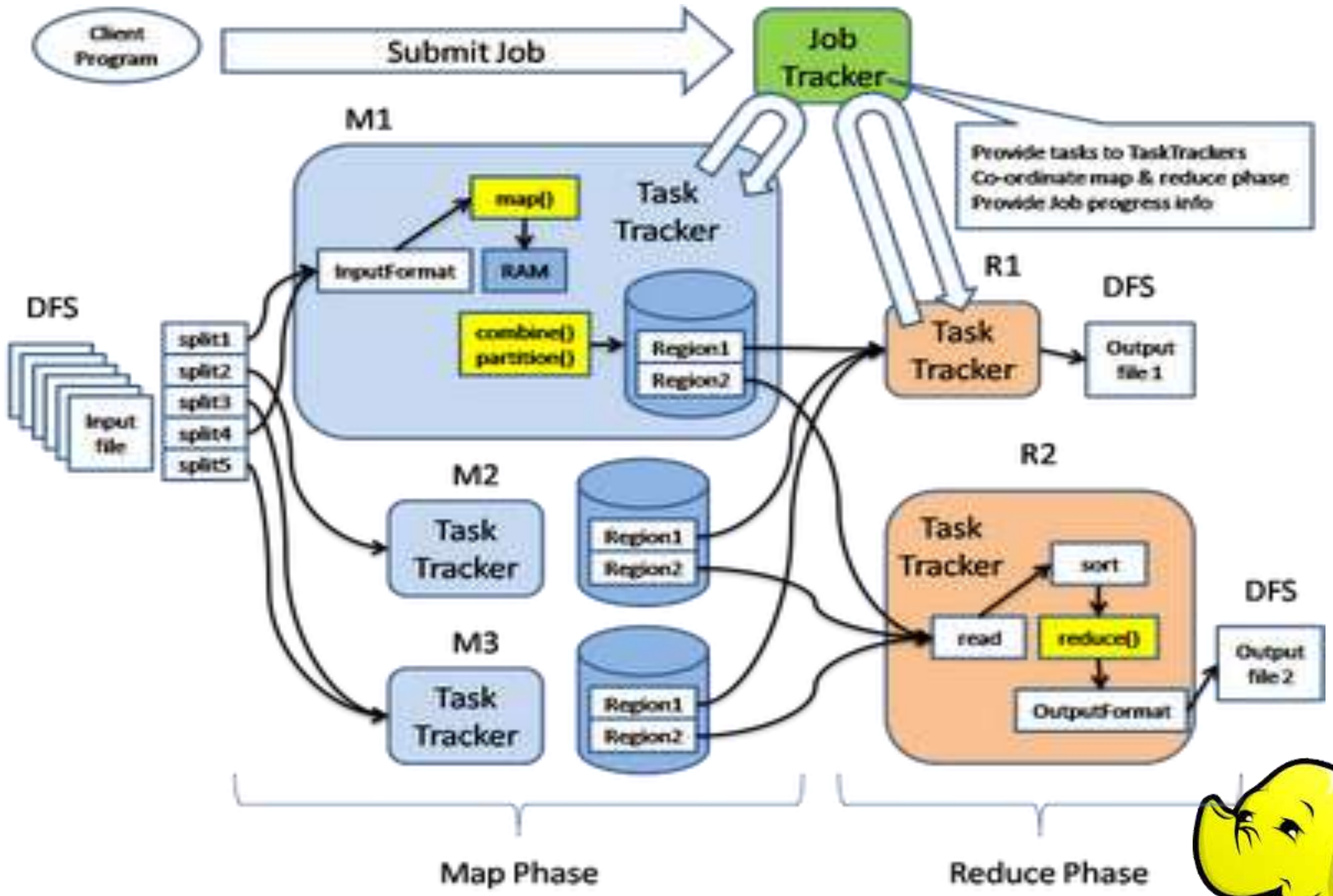
- JobTracker splits up data into smaller tasks("Map") and sends it to the TaskTracker process in each node

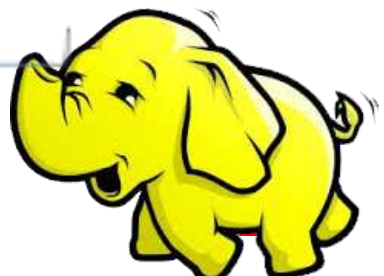


- TaskTracker reports back to the JobTracker node and reports on job progress, sends data ("Reduce") or requests new jobs

MapReduce Engine: JobTracker & TaskTracker

Hadoop's Architecture: MapReduce Engine





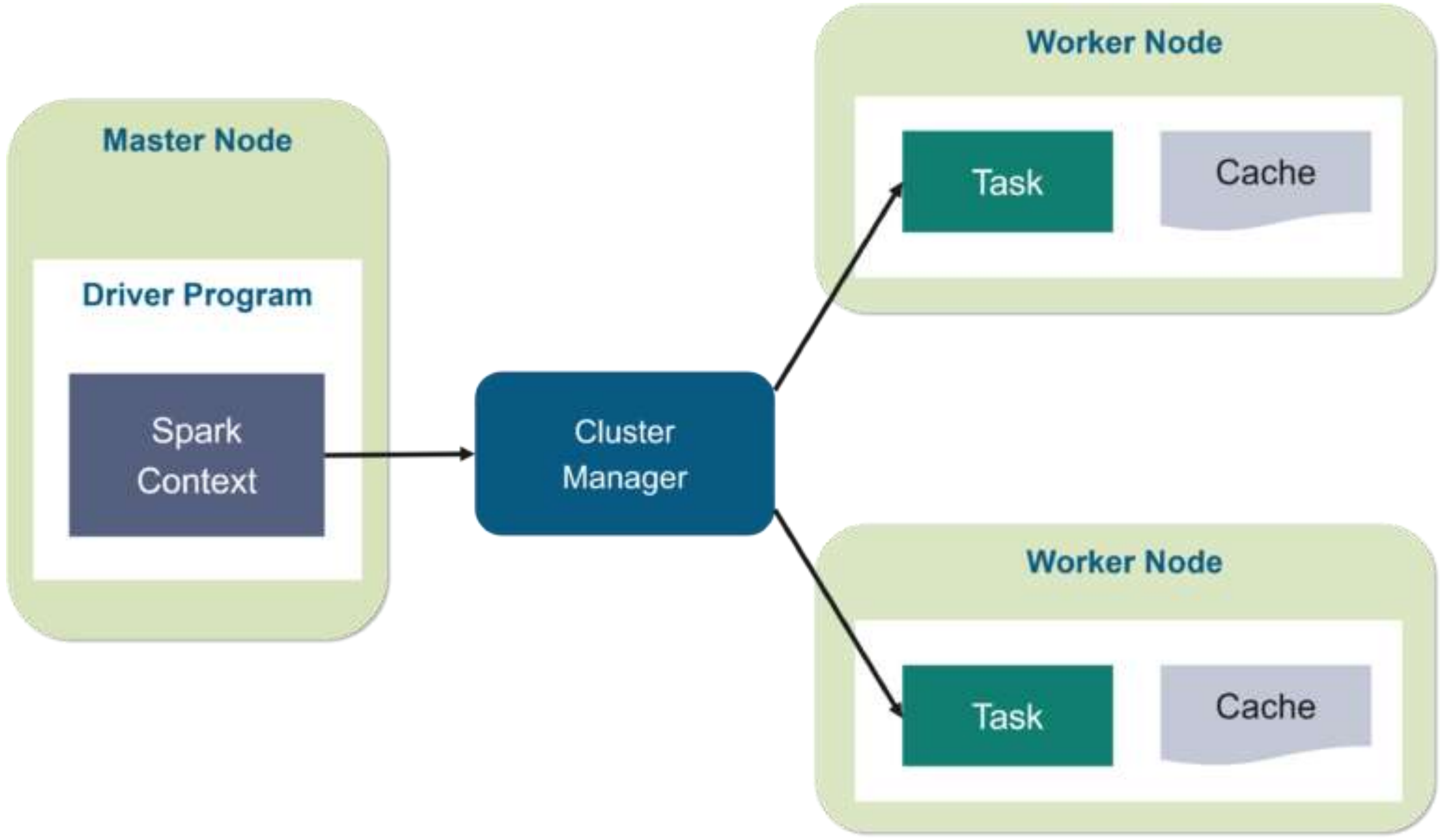
Map/Reduce features

- Java and C++ APIs
  - In Java use Objects, while in C++ bytes
- Each task can process data sets larger than RAM
- Automatic re-execution on failure
  - In a large cluster, some nodes are always slow or flaky
  - Framework re-executes failed tasks
- Locality optimizations
  - Map-Reduce queries HDFS for locations of input data
  - Map tasks are scheduled close to the inputs when possible

Hadoop clusters

- ~20,000 machines running Hadoop
- clusters are 2000 nodes
- Several petabytes of user data (compressed, unreplicated)
- Run hundreds of thousands of jobs every month

Spark Architecture



SPARK

Spark is a cluster – computing framework. It competes with MapReduce and the entire Hadoop ecosystem. Spark doesn't have its own distributed filesystem, but it uses the HDFS.

Polyglot

Spark provides high-level APIs in Java, Scala, Pytl and R. Spark code can be written in any of these languages. It also provides a shell in Scala and Python.

Speed

Spark runs up to 100 times faster than Hadoop MapReduce for large-scale data processing. It is also able to achieve this speed through controlled partitioning.

PowerfulCaching

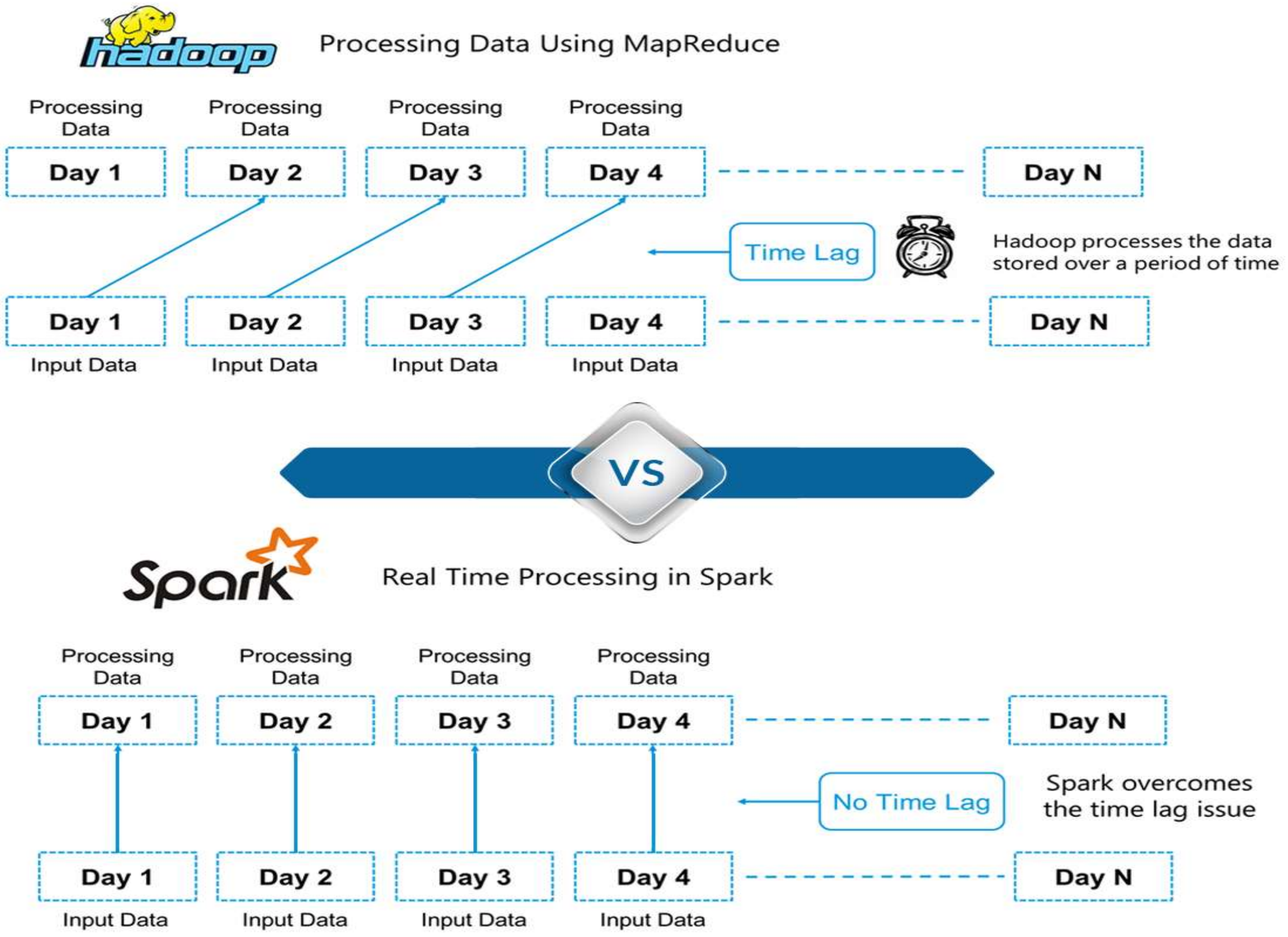
Simple programming layer provides powerful caching and disk persistence capabilities.

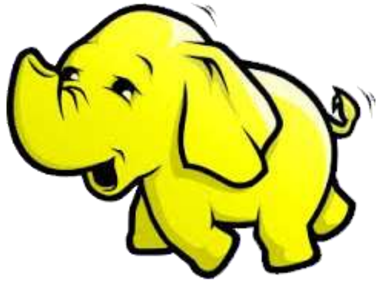
Deployment

It can be deployed through ***Mesos, Hadoop via YARN, or Spark's own cluster manager.***

Real-Time

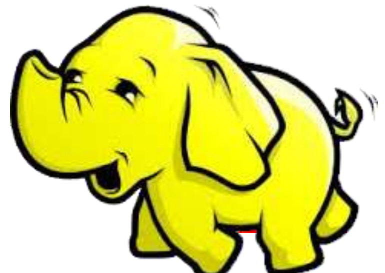
It offers Real-time computation & low latency because of ***in-memory computation.***


Processing Data Using MapReduce
Processing Data
Day 1
Processing Data
Day 2
Processing Data
Day 3
Processing Data
Day 4
Day N
Time Lag
Hadoop processes the data stored over a period of time
Day 1
Input Data
Day 2
Input Data
Day 3
Input Data
Day 4
Input Data
Day N
VS
Spark
Real Time Processing in Spark
Processing Data
Day 1
Processing Data
Day 2
Processing Data
Day 3
Processing Data
Day 4
Day N
No Time Lag
Spark overcomes the time lag issue
Day 1
Input Data
Day 2
Input Data
Day 3
Input Data
Day 4
Input Data
Day N



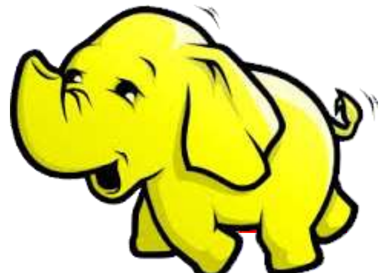
Three main applications of Hadoop:

- Advertisement (Mining user behavior to generate recommendations)
- Searches (group related documents)
- Security (search for uncommon patterns)
    - Hadoop is in use at most organizations that handle big data:
        - Yahoo!
        - Facebook
        - Amazon
        - Netflix
        - Etc…



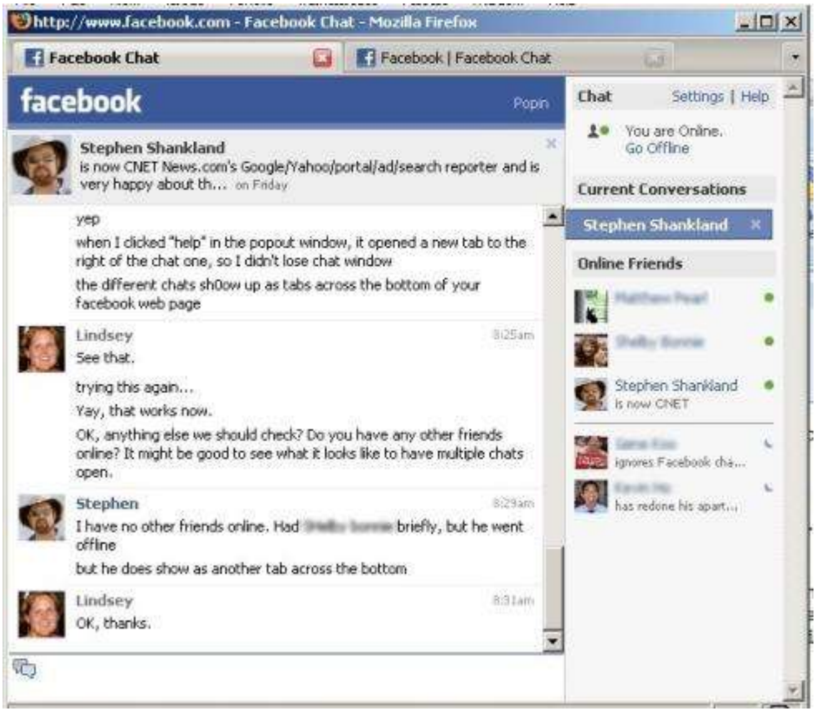
Hadoop in the Wild

- Non-realtime large dataset computing:
  - NY Times was dynamically generating PDFs of articles from 1851-1922
  - Wanted to pre-generate & statically serve articles to improve performance
  - Using Hadoop + MapReduce running on EC2 / S3, converted 4TB of TIFFs into 11 million PDF articles in 24 hrs



Hadoop in the Wild: Facebook Messages

- Design requirements:
  - Integrate display of email, SMS and chat messages between pairs and groups of users
  - Strong control over who users receive messages from
  - Suited for production use between 500 million people immediately after launch
  - Stringent latency & uptime requirements


http://www.facebook.com - Facebook Chat - Mozilla Firefox

Facebook Chat | Facebook | Facebook Chat

facebook Popin

Stephen Shankland
is now CNET News.com's Google/Yahoo/portal/ad/search reporter and is very happy about th... on Friday

yep
when I clicked "help" in the popout window, it opened a new tab to the right of the chat one, so I didn't lose chat window
the different chats sh0ow up as tabs across the bottom of your facebook web page

Lindsey 8:25am
See that.
trying this again...
Yay, that works now.
OK, anything else we should check? Do you have any other friends online? It might be good to see what it looks like to have multiple chats open.

Stephen 8:29am
I have no other friends online. Had [redacted] briefly, but he went offline
but he does show as another tab across the bottom

Lindsey 8:31am
OK, thanks.

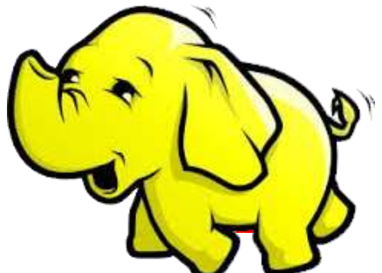
Chat Settings | Help
You are Online.
Go Offline

Current Conversations
Stephen Shankland

Online Friends
[redacted]
[redacted]
Stephen Shankland
is now CNET
[redacted]
ignores Facebook cha...
[redacted]
has redone his apart...

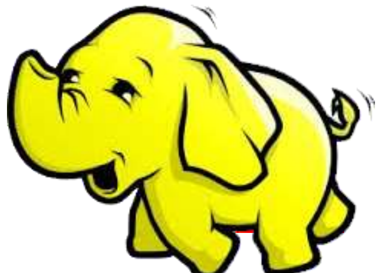
Hadoop in the Wild

- Facebook's solution
  - Hadoop + HBase as foundations
  - Improve & adapt HDFS and HBase to scale to FB's workload and operational considerations
    - Major concern was availability: NameNode is SPOF & failover times are at least 20 minutes
    - Proprietary "AvatarNode": eliminates SPOF, makes HDFS safe to deploy even with 24/7 uptime requirement
    - Performance improvements for realtime workload: RPC timeout. Rather fail fast and try a different DataNode



Hadoop in the Wild

- Classic alternatives
  - These requirements typically met using large MySQL cluster & caching tiers using Memcached
  - Content on HDFS could be loaded into MySQL or Memcached if needed by web tier
- Problems with previous solutions
  - MySQL has low random write throughput… BIG problem for messaging!
  - Difficult to scale MySQL clusters rapidly while maintaining performance
  - MySQL clusters have high management overhead, require more expensive hardware



Hadoop Highlights

- Distributed File System
- Fault Tolerance
- Open Data Format
- Flexible Schema
- Queryable Database

Why use Hadoop?

- Need to process Multi Petabyte Datasets
- Data may not have strict schema
- Expensive to build reliability in each application
- Nodes fails everyday
- Need common infrastructure
- Very Large Distributed File System
- Assumes Commodity Hardware
- Optimized for Batch Processing
- Runs on heterogeneous OS