

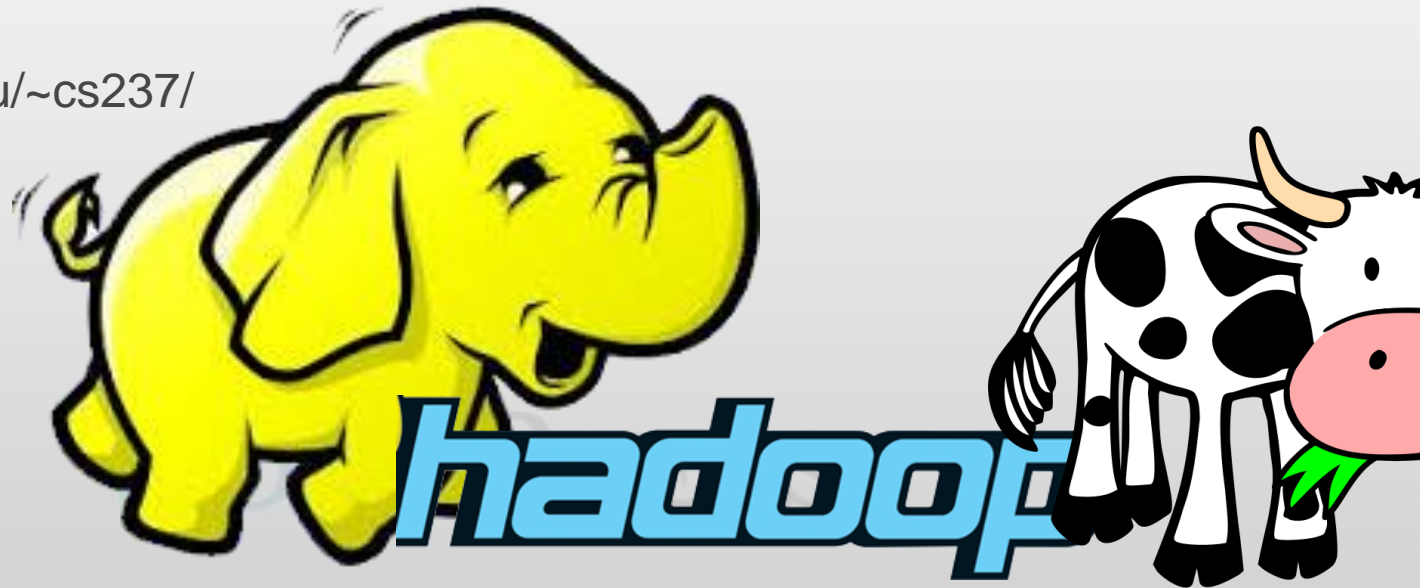
```
<?php
/**
 * @package WordPress
 * @subpackage Default_Theme
 */
?>
<!DOCTYPE html PUBLIC "-//W3C//
html xmlns="http://www.w3.
.org/1999/xhtml"
>
<head profile="http://
www.w3.org/TR/xhtml-
1/DTD/xhtml1-strict-1.1.
1.dtd" >
<meta http-equiv="Con
tent-Type" content="text/
html; charset=utf-8" />
<title><?php vp
</?php>
</title>
<link rel="stylesheet"
type="text/css" href="
css/all.css" />
</head>
<body>
</body>
</html>
```

Hadoop, a distributed framework for Big Data

Move aside cows!
It's time for the BIG guys

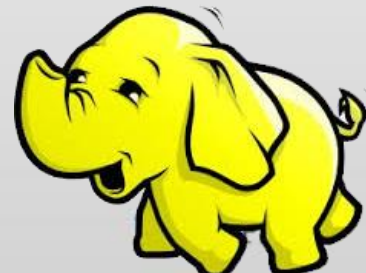
Slides and graphics borrow heavily from **Prof. Nalini Venkatasubramanian**

<http://www.ics.uci.edu/~cs237/>

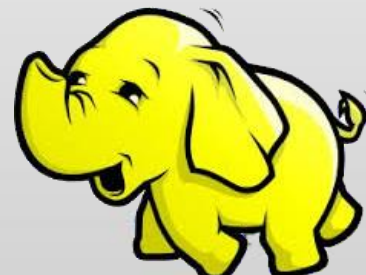


BIG Data, how big is BIG?

- Not about size, but how data is managed
- Relational databases was all about *organizing* data into tables
- Sometimes it is just too time consuming, or the data is just too big, to organize it in order to do simple queries
- Much data is unstructured or semi-structured and we'd like to process it in parallel
- Data warehouses



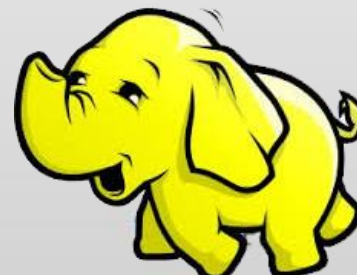
- 1. Introduction: Hadoop's history and advantages**
- 2. Architecture in detail**
- 3. Hadoop in industry**



```
<!--  
 * @package WordPress  
 * @subpackage Default_Theme  
 *-->  
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" profile="http://www.w3.org/2002/05/xhtml-profile-1.0">  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
<title></?php echo $page_title ?></title>  
<link rel="stylesheet" type="text/css" href="http://www.wordpress.org/wp-content/themes/default/css/style.css" />  
</html>
```

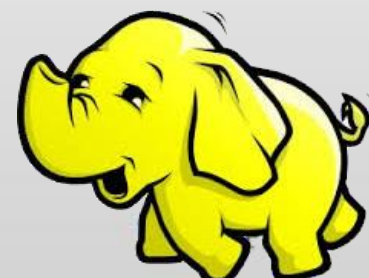


- Open-source implementation of a Map-Reduce framework for reliable, scalable, distributed computing and data storage.
- It is a flexible architecture for large scale computation and data processing on a network of commodity hardware.



Brief History of Hadoop

- Designed to answer the question:
“How to process big data with reasonable cost and time?”



Search engines in 1990s



MetaCrawler Parallel Web Search Service

by [Erik Selberg](#) and [Oren Etzioni](#)

Try the new [MetaCrawler Beta!](#)
If you're searching for a person's home page, try [Abov!](#)

• [Examples](#) • [Beta Site](#) • [Add Site](#) • [About](#) •

Search for:
 as a Phrase All of these words Any of these words

For better results, please specify:
Search Region: Search Sites:

Performance parameters:
Max wait: minutes Match type:

[[About](#) | [Help](#) | [Problems](#) | [Add Site](#) | [Search](#)]
webmaster@metacrawler.com
© Copyright 1995, 1996 Erik Selberg and Oren Etzioni

1996

The screenshot shows the Excite search engine homepage. At the top, there are navigation links for "excite home", "maps", "news", and "people finder". Below these are icons for "search", "reviews", "city.net", "new live", and "reference?". The main search area features a "What:" search box with a red "search" button, a "Where:" dropdown menu set to "World Wide Web", and a "Help" link. Below the search area, there are several promotional banners: "Excite Search: twice the power of the competition.", "Excite Reviews: site reviews by the web's best editorial team.", and "Excite Seeing Tours." A sidebar on the left lists various categories like "Arts", "Business", "Computing", "Education", "Entertainment", "Health", "Hobbies", "Life & Style", "Money", "News & Reference", "Personal Pages", "Politics & Law", "Regional", "Science", "Shopping", and "Sports".

1996

Serious Sports Fans Only \$1,000,000 in Cash and Prizes!
For serious sports fans only! Play Fantasy Football!



It's amazing where
Go Get It will get you.

Find:

[Enhance your search.](#)



[New Search](#) • [TopNews](#) • [Sites by Subject](#) • [Top 5% Sites](#) • [City Guide](#) • [Pictures & Sounds](#)
[PeopleFind](#) • [Point Review](#) • [Road Maps](#) • [Software](#) • [About Lycos](#) • [Club Lycos](#) • [Help](#)

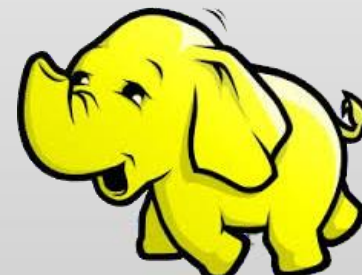
[Add Your Site to Lycos](#)

Copyright © 1996 Lycos™, Inc. All Rights Reserved.
Lycos is a trademark of Carnegie Mellon University.
[Questions & Comments](#)

1996

The screenshot shows the WRED Search Center interface. At the top, there are navigation links for "HELP", "WIRED NEWS", "NOTWIRED", "WIRED MAGAZINE", and "SUCK.COM". The main search area features a "look for:" search box with a dropdown menu set to "all the words" and a red "SEARCH" button. Below the search area, there are several filters: "Date" set to "in the last week", "Country" set to "North America (.com)", and "Include media type" checkboxes for "Image", "Audio", "Video", and "Downloads". The "Return Results:" section shows "10" results and "full descriptions". A sidebar on the right contains links for "Sandbox Entertainment", "Shop WIRED Holiday Gift Guide", "SOMETHING HAS SURVIVED.", "Log", "Cyberlan Outpost", "Microsoft® Expedia™ Travel", and "ONSALE".

1997



Google search engines

```
<!-- Package WordPress -->
<!-- Package Default_Theme -->
</?
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head profile="http://gmpg.org/WordPress-2.0-profile/" >
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title></?php wp_title()></title>
<link rel="stylesheet" type="text/css" href="http://www.example.com/wp-content/themes/default/style.css" />
<link rel="stylesheet" type="text/css" href="http://www.example.com/wp-content/themes/default/images.css" />
</head>
<body >
</body>
</html>
```



1998



2003 The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google*



MapReduce: Simplified Data Processing on Large Clusters

2004

Jeffrey Dean and Sanjay Ghemawat
jeff@google.com, sanjay@google.com
Google, Inc.



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,wilson,hkerr,m3b,tsihar,files,gruber}@google.com
Google, Inc.



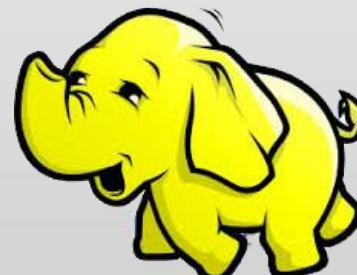
2006

Abstract

Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large number of nodes and petabytes of data across thousands of commodity servers. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google File Service. These applications place very different demands on Bigtable, both in terms of data size (from URLs to gigabytes to satellite imagery) and latency requirements.

Bigtable achieves scalability and high performance, but Bigtable provides a different interface than such systems. Bigtable does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format, and allows clients to reason about the locality properties of data represented in the underlying storage. Data is indexed using row and column names that can be arbitrary strings. Bigtable also treats data as uninterpreted strings.

2016



Hadoop's Developers



Doug Cutting



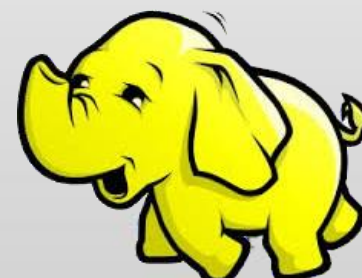
2005: Doug Cutting and Michael J. Cafarella developed Hadoop to support distribution for the [Nutch](#) search engine project.



The project was funded by Yahoo.



2006: Yahoo gave the project to Apache Software Foundation.



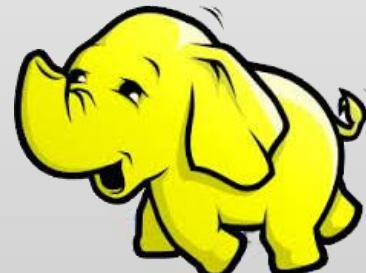
What is Hadoop?

- **Hadoop:**

- An open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license.

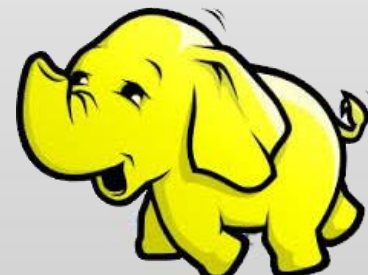
- **Goals / Requirements:**

- Data and Processing abstractions facilitate queries of large, dynamic, and 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



Hadoop's Architecture

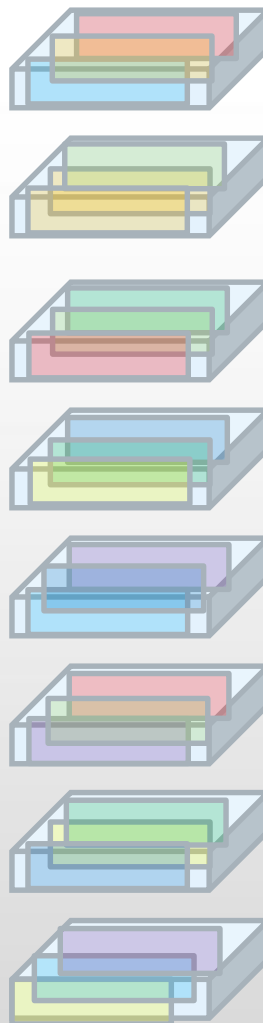
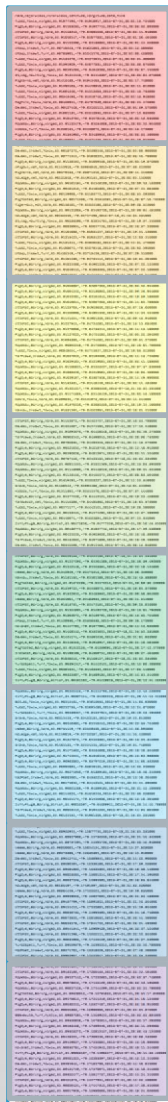
- Distributed, with some *modest* centralization
- Main nodes of cluster are where most of the computational power and storage of the system lies
- Main nodes run TaskTracker to accept and reply to MapReduce tasks, and also DataNode to store needed blocks closely as possible
- Central control node runs NameNode to keep track of HDFS directories & files, and JobTracker to dispatch compute tasks to TaskTracker
- Written in Java, also supports Python and Ruby



Hadoop's Data Model

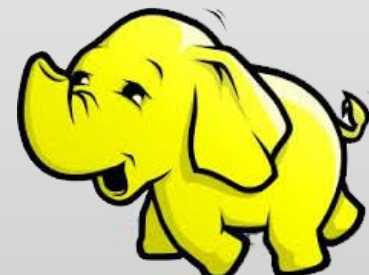
1. Given giant files
2. Chops them up into good-sized chunks (64Mb)
3. Replicate and Distribute them

Hadoop's Distributed File System



Each chunk is replicated 3 times, and placed on a different processing node

A name server (actually 2) keeps track of where the chunks are



Hadoop's Processing Model

Whenever we query the dataset, its done in the following stages:

Map:

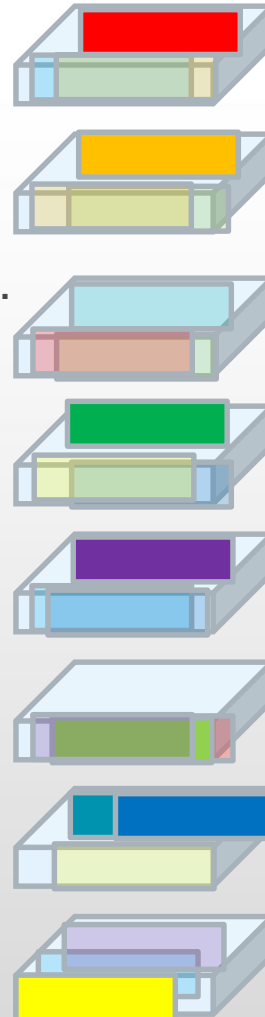
1. A processor is assigned to each chunk.
2. That processor scans, filters, and maps each data item into key-value pairs.
3. Keys are locally binned

Shuffle:

4. Bins with common keys are consolidated by broadcasting them to a common node

Reduce:

5. Final processing is done of within each bin, often agglomerative-like operations

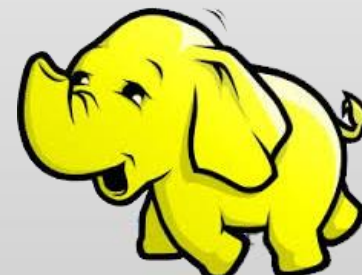
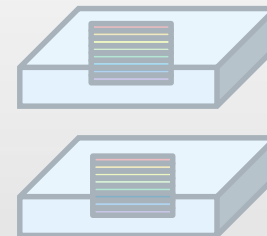


MapReduce

Distributed processing

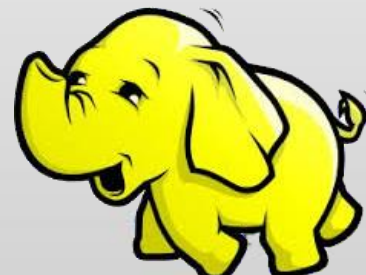
Generally balanced, but no guarantees

Processing occurs at the data source

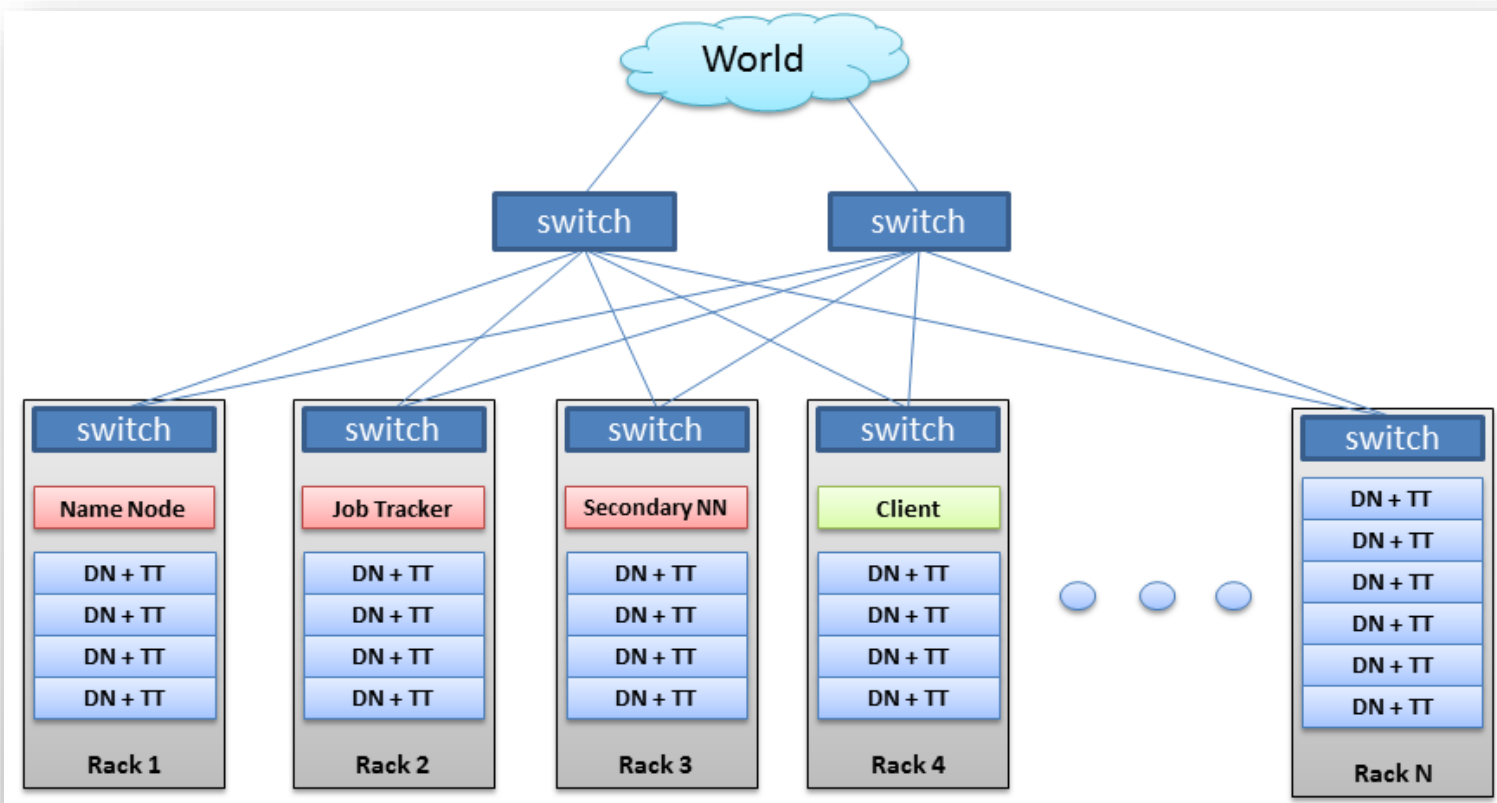


Hadoop's Architecture

- Hadoop Distributed FileSystem (Chops up and distributes data)
- Tailored to needs of MapReduce
- Targeted towards many reads of file streams
- **Writes** are more costly
- High degree of data replication (3x by default)
- No need for RAID on normal nodes
- Large blocksize (64MB, bigger than database pages)
- Location awareness of DataNodes in network

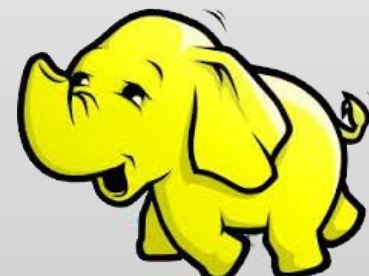


Hadoop's Reality



Also need to keep track of:

1. Where the data chunks are
2. What the state of multiple MapReduce jobs are in
3. Redundancy in case there are either H/W or network issues

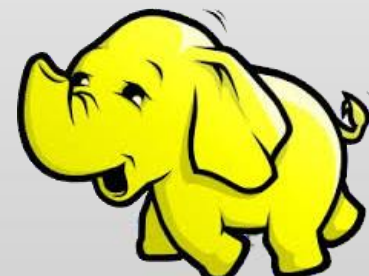


Hadoop's Architecture



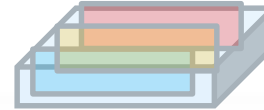
NameNode:

- Stores metadata for the files, like the directory structure of a typical FS.
- The server holding the NameNode instance is quite crucial, so we keep a replicate.
- Transaction log for file deletes/adds, etc. Does not use transactions for whole blocks or file-streams, only metadata.
- Handles creation of more replica blocks when necessary after a DataNode failure



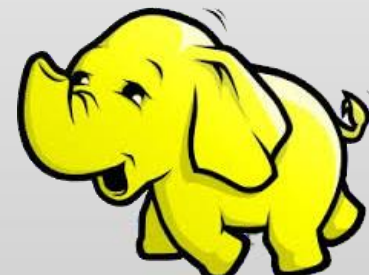
Hadoop's Architecture

```
<?xml version="1.0" ?>
<?package WordPress
<?subpackage Default_Theme
</?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" class="no-js">
<head profile="http://gmpg.org/xfn/11">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title><?php wp_title( ' ', true, true );></title>
<link rel="stylesheet" type="text/css" href="http://www.example.com/wp-content/themes/Default/css/style.css">
<link rel="stylesheet" type="text/css" href="http://www.example.com/wp-content/themes/Default/css/fonts.css">
<script></script>
</head>
<body>
</body>
</html>
```

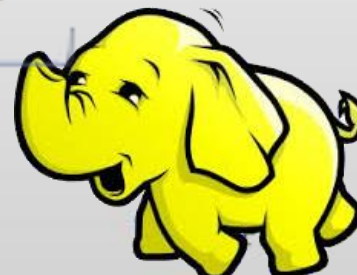
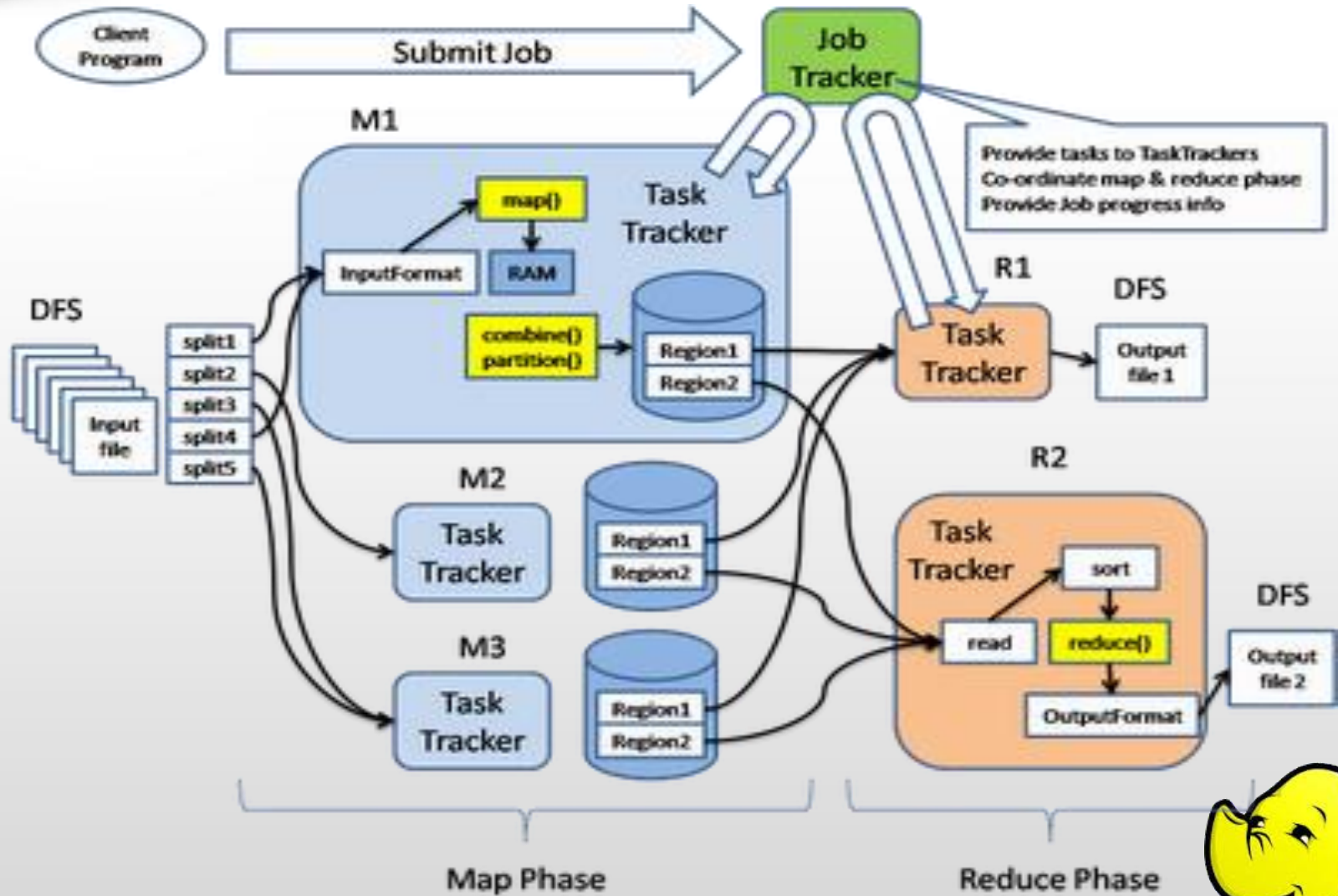


DataNode:

- Stores the actual data in HDFS
- Can run on any underlying filesystem (ext3/4, NTFS, etc)
- NameNode decides and tracks which blocks it has
- NameNode replicates blocks 3x
- Don't need to Homogenous
 - Different levels of performance
 - Different operating systems



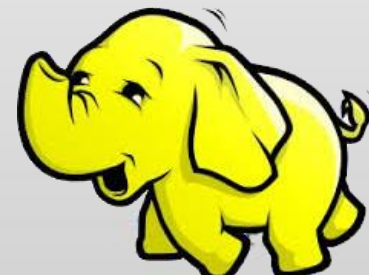
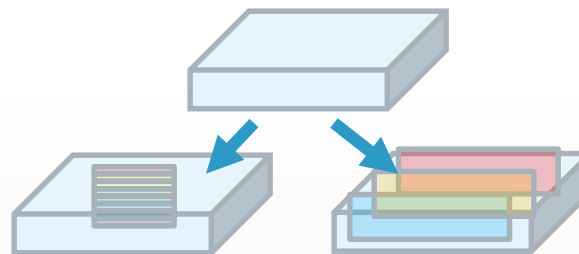
Job-Tracker has a key role in the MapReduce Engine



Hadoop's Architecture

MapReduce Engine:

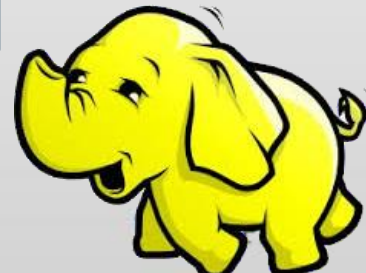
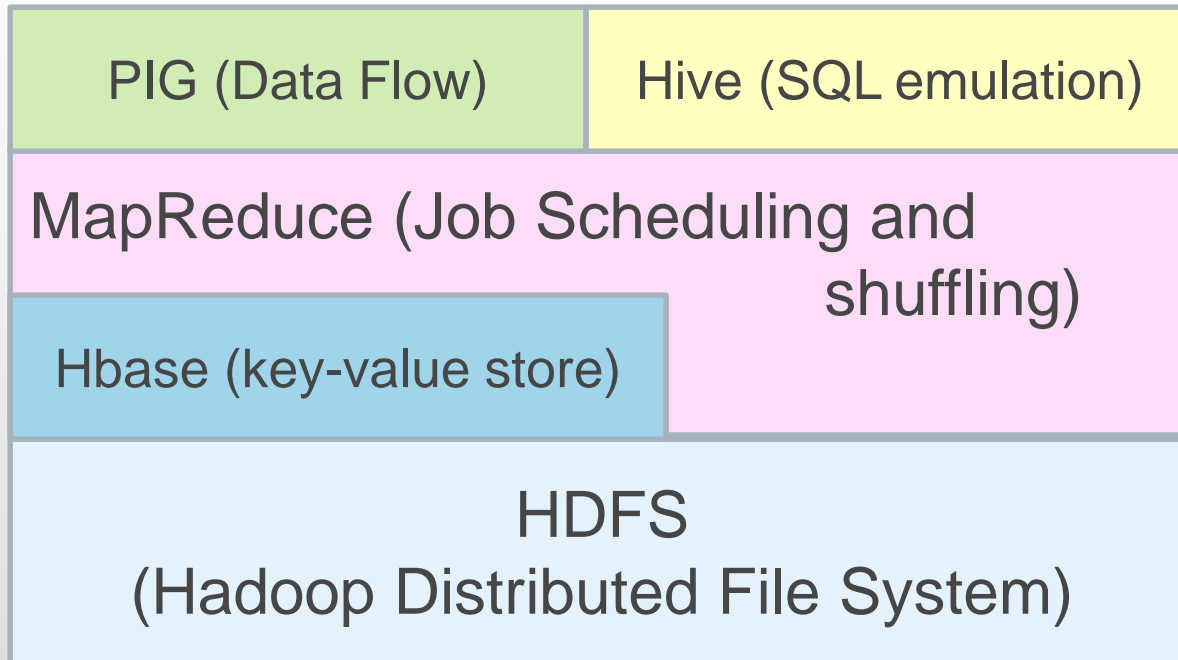
- JobTracker & TaskTracker
- JobTracker splits up data into smaller tasks (“Map”) and sends it to the TaskTracker process in each node
- TaskTracker reports back to the JobTracker node and reports on job progress, sends data (“Reduce”) or requests new jobs
- You can have multiple of these, but only one is responsible for a given query



Hadoop Layer Cake

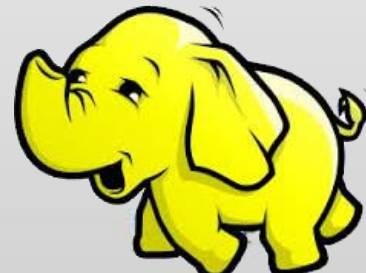
Most interaction with Hadoop is mediated by job managers using high-level APIs

1. PIG, a scripting language, with FOREACH, GROUP, FILTER, and ORDER constructs
2. Hive, SQL syntax, declarative specification



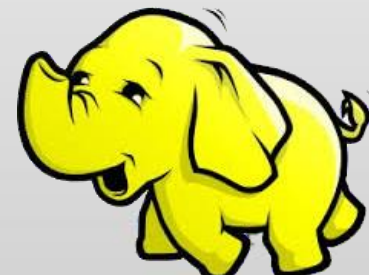
Hadoop in the Wild

- Hadoop is in use at most organizations that handle big data:
 - Yahoo!
 - Facebook
 - Amazon
 - Netflix
 - Etc...
- Some examples of scale:
 - Yahoo!'s Search Webmap runs on 10,000 core Linux cluster and powers Yahoo! Web search
 - FB's Hadoop cluster hosts 100+ PB of data (July, 2012) & growing at ½ PB/day (Nov, 2012)



Hadoop in the Wild

- System requirements
 - High write throughput
 - Cheap, elastic storage
 - Low latency
 - High consistency (within a single data center good enough)
 - Disk-efficient sequential and random read performance



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

