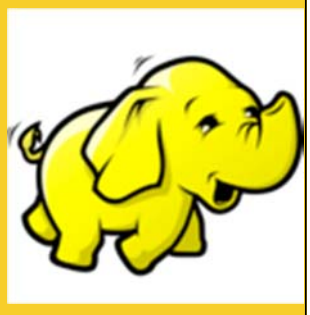
Hadoop Introduction

Originals of Slides and Source Code for Examples:
http://www.coreservlets.com/hadoop-tutorial/

© 2012 coreservlets.com and Dima May



Agenda

- Big Data
- Hadoop Introduction
- History
- Comparison to Relational Databases
- Hadoop Eco-System and Distributions
- Resources

Big Data

- Information Data Corporation (IDC) estimates data created in 2010 to be

  1.2 ZETTABYTES

  (1.2 Trillion Gigabytes)

- Companies continue to generate large amounts of data, here are some 2011 stats:
  - Facebook ~ 6 billion messages per day
  - EBay ~ 2 billion page views a day, ~ 9 Petabytes of storage
  - Satellite Images by Skybox Imaging ~ 1 Terabyte per day

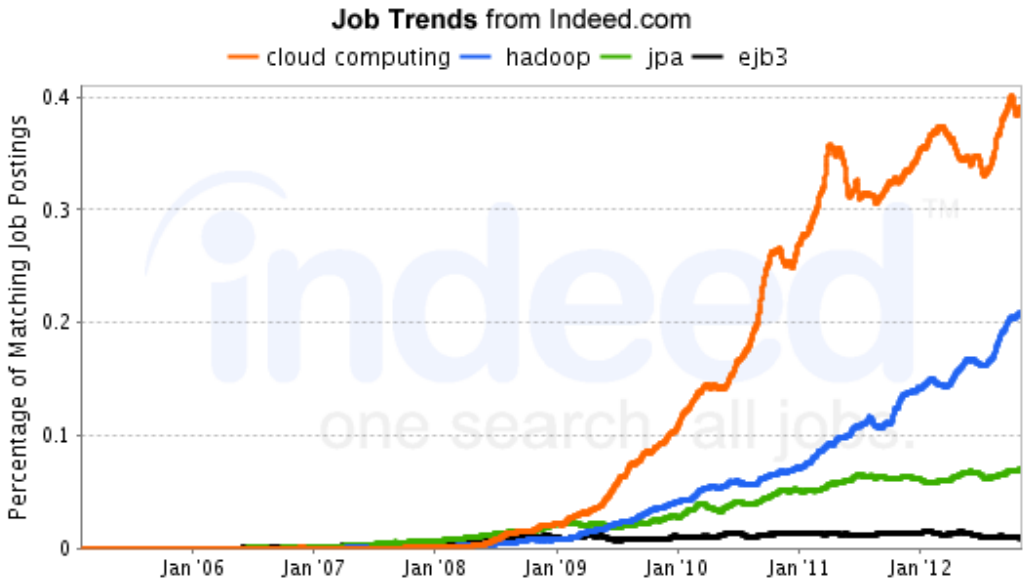
Sources:
"Digital Universe" study by IDC; http://www.emc.com/leadership/programs/digital-universe.htm
Hadoop World 2011 Keynote: Hugh E. Williams, eBay
Hadoop World 2011: Building Realtime Big Data Services at Facebook with Hadoop and HBase
Hadoop World 2011: Indexing the Earth – Large Scale Satellite Image Processing Using Hadoop

Hadoop

- **Existing tools were not designed to handle such large amounts of data**

- **"The Apache™ Hadoop™ project develops open-source software for reliable, scalable, distributed computing." - http://hadoop.apache.org**
  - Process Big Data on clusters of commodity hardware
  - Vibrant open-source community
  - Many products and tools reside on top of Hadoop

Hadoop Jobs



Source: http://www.indeed.com/jobanalytics/jobtrends?q=cloud+computing%2C+hadoop%2C+jpa%2C+ejb3&l=

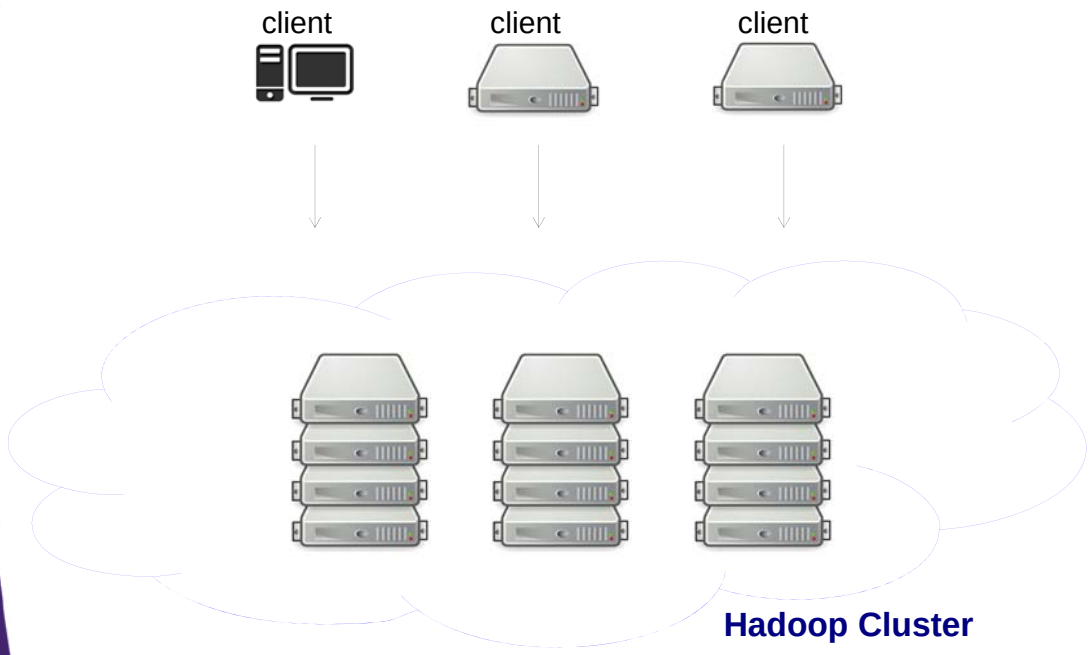
Who Uses Hadoop?

Source: http://wiki.apache.org/hadoop/PoweredBy

Data Storage

- **Storage capacity has grown exponentially but read speed has not kept up**
  - 1990:
    - Store 1,400 MB
    - Transfer speed of 4.5MB/s
    - Read the entire drive in ~ 5 minutes
  - 2010:
    - Store 1 TB
    - Transfer speed of 100MB/s
    - Read the entire drive in ~ 3 hours
- **Hadoop - 100 drives working at the same time can read 1TB of data in 2 minutes**

Hadoop Cluster



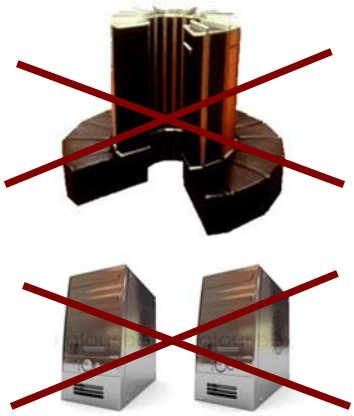
Hadoop Cluster

- **A set of "cheap" commodity hardware**
- **Networked together**
- **Resides in the same location**
  - Set of servers in a set of racks in a data center

Use Commodity Hardware

- **"Cheap" Commodity Server Hardware**
  - No need for super-computers, use commodity unreliable hardware
  - Not desktops

NOT



BUT

Hadoop System Principles

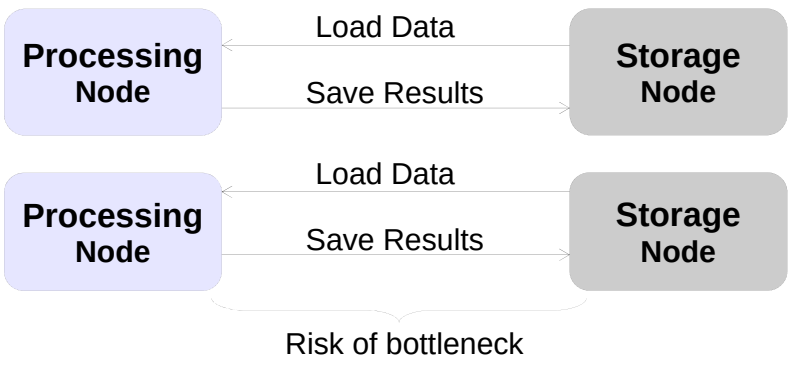
- **Scale-Out rather than Scale-Up**
- **Bring code to data rather than data to code**
- **Deal with failures – they are common**
- **Abstract complexity of distributed and concurrent applications**

Scale-Out Instead of Scale-Up

- **It is harder and more expensive to scale-up**
  - Add additional resources to an existing node (CPU, RAM)
  - Moore's Law can't keep up with data growth
  - New units must be purchased if required resources can not be added
  - Also known as scale vertically
- **Scale-Out**
  - Add more nodes/machines to an existing distributed application
  - Software Layer is designed for node additions or removal
  - Hadoop takes this approach - A set of nodes are bonded together as a single distributed system
  - Very easy to scale down as well

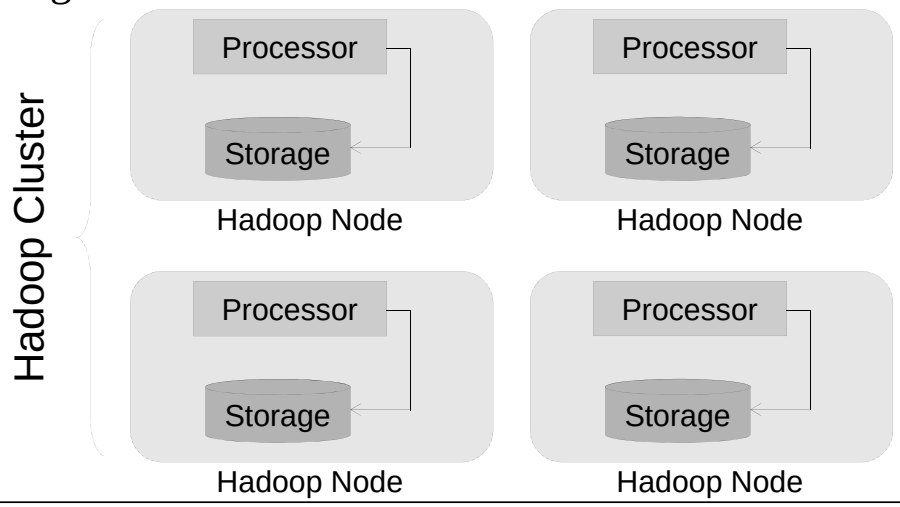
Code to Data

- **Traditional data processing architecture**
  - nodes are broken up into separate processing and storage nodes connected by high-capacity link
  - Many data-intensive applications are not CPU demanding causing bottlenecks in network



Code to Data

- **Hadoop co-locates processors and storage**
  - Code is moved to data (size is tiny, usually in KBs)
  - Processors execute code and access underlying local storage



Failures are Common

- **Given a large number machines, failures are common**
  - Large warehouses may see machine failures weekly or even daily
- **Hadoop is designed to cope with node failures**
  - Data is replicated
  - Tasks are retried

Abstract Complexity

- **Hadoop abstracts many complexities in distributed and concurrent applications**
  - Defines small number of components
  - Provides simple and well defined interfaces of interactions between these components
- **Frees developer from worrying about system-level challenges**
  - race conditions, data starvation
  - processing pipelines, data partitioning, code distribution
  - etc.
- **Allows developers to focus on application development and business logic**

History of Hadoop

- **Started as a sub-project of Apache Nutch**
  - Nutch's job is to index the web and expose it for searching
  - Open Source alternative to Google
  - Started by Doug Cutting
- **In 2004 Google publishes Google File System (GFS) and MapReduce framework papers**
- **Doug Cutting and Nutch team implemented Google's frameworks in Nutch**
- **In 2006 Yahoo! hires Doug Cutting to work on Hadoop with a dedicated team**
- **In 2008 Hadoop became Apache Top Level Project**
  - http://hadoop.apache.org

Naming Conventions?

- **Doug Cutting drew inspiration from his family**
  - Lucene: Doug's wife's middle name
  - Nutch: A word for "meal" that his son used as a toddler
  - Hadoop: Yellow stuffed elephant named by his son

Comparisons to RDBMS

- **Until recently many applications utilized Relational Database Management Systems (RDBMS) for batch processing**
  - Oracle, Sybase, MySQL, Microsoft SQL Server, etc.
  - Hadoop doesn't fully replace relational products; many architectures would benefit from both Hadoop and a Relational product(s)
- **Scale-Out vs. Scale-Up**
  - RDBMS products scale up
    - Expensive to scale for larger installations
    - Hits a ceiling when storage reaches 100s of terabytes
  - Hadoop clusters can scale-out to 100s of machines and to petabytes of storage

Comparisons to RDBMS (Continued)

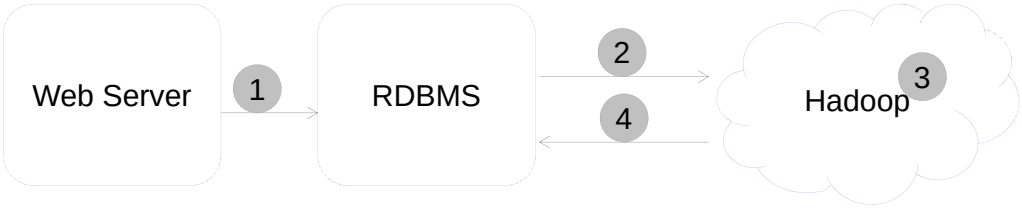
- **Structured Relational vs. Semi-Structured vs. Unstructured**
  - RDBMS works well for structured data - tables that conform to a predefined schema
  - Hadoop works best on Semi-structured and Unstructured data
    - Semi-structured may have a schema that is loosely followed
    - Unstructured data has no structure whatsoever and is usually just blocks of text (or for example images)
    - At processing time types for key and values are chosen by the implementer
  - Certain types of input data will not easily fit into Relational Schema such as images, JSON, XML, etc...

Comparison to RDBMS

- **Offline batch vs. online transactions**
  - Hadoop was not designed for real-time or low latency queries
  - Products that do provide low latency queries such as HBase have limited query functionality
  - Hadoop performs best for offline batch processing on large amounts of data
  - RDBMS is best for online transactions and low-latency queries
  - Hadoop is designed to stream large files and large amounts of data
  - RDBMS works best with small records

Comparison to RDBMS

- **Hadoop and RDBMS frequently complement each other within an architecture**
- **For example, a website that**
  - has a small number of users
  - produces a large amount of audit logs



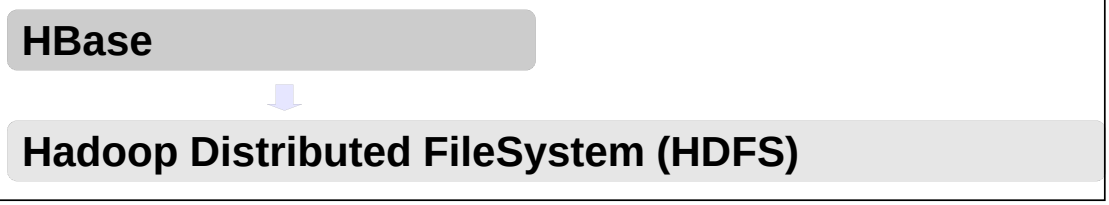
1. Utilize RDBMS to provide rich User Interface and enforce data integrity
2. RDBMS generates large amounts of audit logs; the logs are moved periodically to the Hadoop cluster
3. All logs are kept in Hadoop; Various analytics are executed periodically
4. Results copied to RDBMS to be used by Web Server; for example "suggestions" based on audit history

Hadoop Eco System

- **At first Hadoop was mainly known for two core products:**
  - HDFS: Hadoop Distributed FileSystem
  - MapReduce: Distributed data processing framework
- **Today, in addition to HDFS and MapReduce, the term also represents a multitude of products:**
  - HBase: Hadoop column database; supports batch and random reads and limited queries
  - Zookeeper: Highly-Available Coordination Service
  - Oozie: Hadoop workflow scheduler and manager
  - Pig: Data processing language and execution environment
  - Hive: Data warehouse with SQL interface

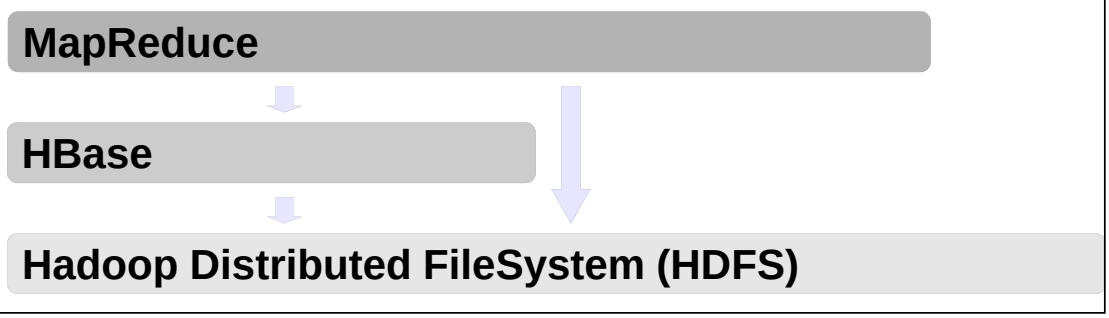
Hadoop Eco System

- **To start building an application, you need a file system**
  - In Hadoop world that would be Hadoop Distributed File System (HDFS)
  - In Linux it could be ext3 or ext4
- **Addition of a data store would provide a nicer interface to store and manage your data**
  - HBase: A key-value store implemented on top of HDFS
  - Traditionally one could use RDBMS on top of a local file system



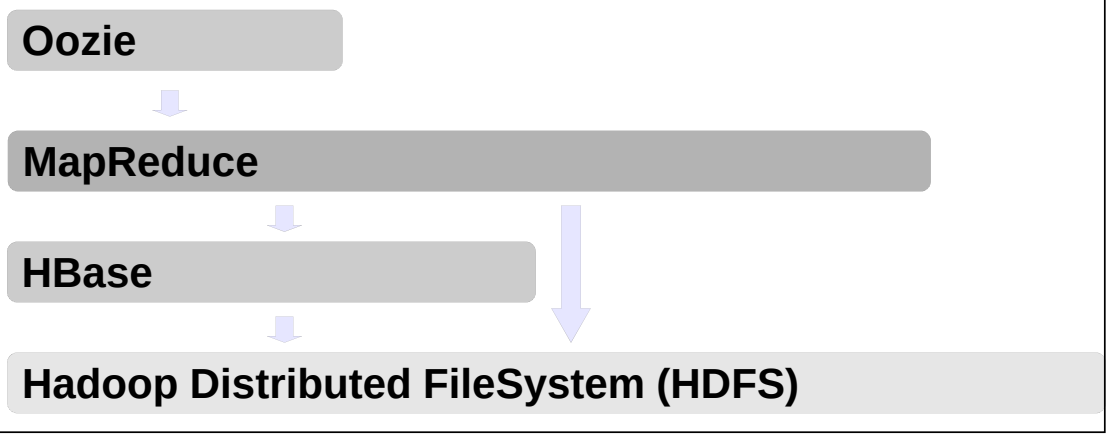
Hadoop Eco System

- **For batch processing, you will need to utilize a framework**
  - In Hadoop's world that would be MapReduce
  - MapReduce will ease implementation of distributed applications that will run on a cluster of commodity hardware



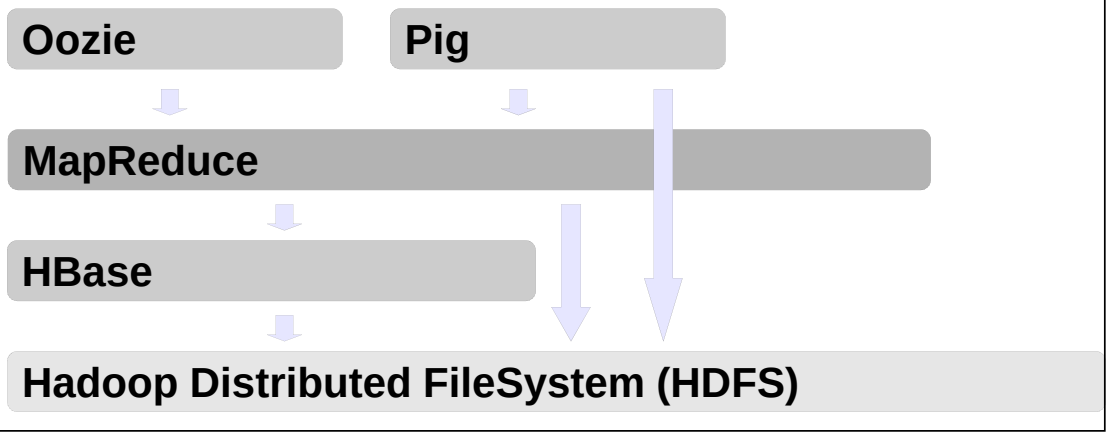
Hadoop Eco System

- **Many problems lend themselves to a MapReduce solution with multiple jobs**
  - Apache Oozie is a popular MapReduce workflow and coordination product



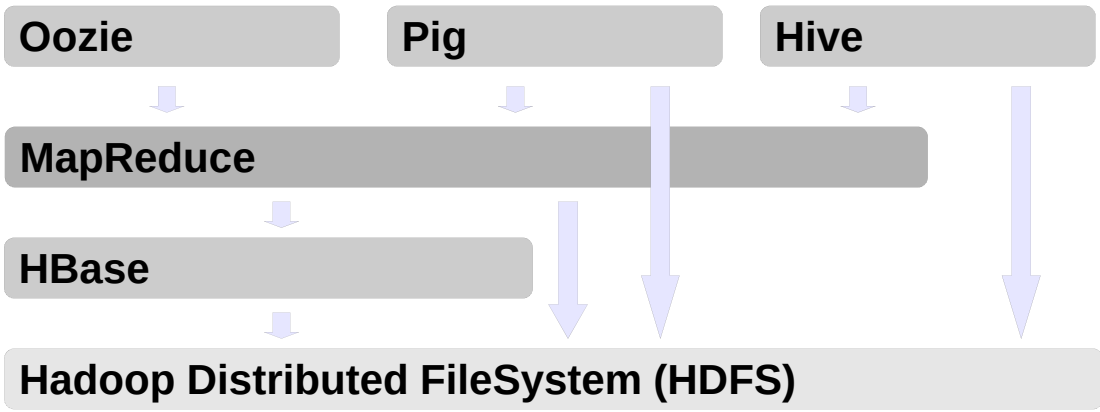
Hadoop Eco System

- **MapReduce paradigm may not work well for analysts and data scientists**
  - Addition of Apache Pig, a high-level data flow scripting language, may be beneficial



Hadoop Eco System

- **Your organization may have a good number of SQL experts**
  - Addition of Apache Hive, a data warehouse solution that provides a SQL based interface, may bridge the gap



Hadoop Distributions

- **Let's say you go download Hadoop's HDFS and MapReduce from http://hadoop.apache.org/**
- **At first it works great but then you decide to start using HBase**
  - No problem, just download HBase from http://hadoop.apache.org/ and point it to your existing HDFS installation
  - But you find that HBase can only work with a previous version of HDFS, so you go downgrade HDFS and everything still works great
- **Later on you decide to add Pig**
  - Unfortunately the version of Pig doesn't work with the version of HDFS, it wants you to upgrade
  - But if you upgrade you'll break HBase...

Hadoop Distributions

- **Hadoop Distributions aim to resolve version incompatibilities**
- **Distribution Vendor will**
  - Integration Test a set of Hadoop products
  - Package Hadoop products in various installation formats
    - Linux Packages, tarballs, etc.
  - Distributions may provide additional scripts to execute Hadoop
  - Some vendors may choose to backport features and bug fixes made by Apache
  - Typically vendors will employ Hadoop committers so the bugs they find will make it into Apache's repository

Distribution Vendors

- **Cloudera Distribution for Hadoop (CDH)**
- **MapR Distribution**
- **Hortonworks Data Platform (HDP)**
- **Apache BigTop Distribution**
- **Greenplum HD Data Computing Appliance**

Cloudera Distribution for Hadoop (CDH)

- **Cloudera has taken the lead on providing Hadoop Distribution**
  - Cloudera is affecting the Hadoop eco-system in the same way RedHat popularized Linux in the enterprise circles
- **Most popular distribution**
  - http://www.cloudera.com/hadoop
  - 100% open-source
- **Cloudera employs a large percentage of core Hadoop committers**
- **CDH is provided in various formats**
  - Linux Packages, Virtual Machine Images, and Tarballs

Cloudera Distribution for Hadoop (CDH)

- **Integrates majority of popular Hadoop products**
  - HDFS, MapReduce, HBase, Hive, Mahout, Oozie, Pig, Sqoop, Whirr, Zookeeper, Flume
- **CDH4 is used in this class**

Supported Operating Systems

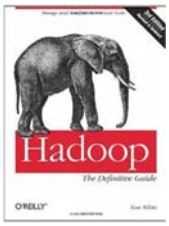
- **Each Distribution will support its own list of Operating Systems (OS)**
- **Common OS supported**
  - Red Hat Enterprise
  - CentOS
  - Oracle Linux
  - Ubuntu
  - SUSE Linux Enterprise Server

- **Please see vendors documentation for supported OS and version**
  - Supported Operating Systems for CDH4: https://ccp.cloudera.com/display/CDH4DOC/Before+You+Install+CDH4+on+a+Cluster#BeforeYouInstallCDH4onaCluster-SupportedOperatingSystemsforCDH4

Resources

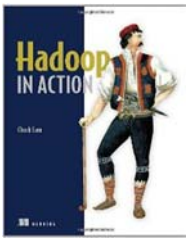
- **Apache Hadoop Documentation**
  - http://hadoop.apache.org
- **Each project will have their own documentation artifacts and usually a wiki**
- **Each Hadoop Distribution Vendor provides documentation as well:**
  - For example: https://ccp.cloudera.com/display/DOC/Documentation

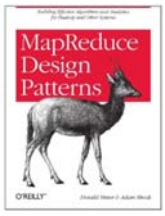
Resources: Books



Hadoop: The Definitive Guide
Tom White (Author)
O'Reilly Media; 3rd Edition (May6, 2012)

Hadoop in Action
Chuck Lam (Author)
Manning Publications; 1st Edition (December, 2010)



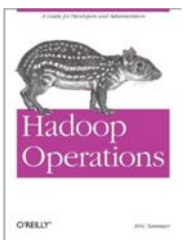


MapReduce Design Patterns
Donald Miner (Author), Adam Shook (Author)
O'Reilly Media (November 22, 2012)

Resources: Books

HBase: The Definitive Guide
Lars George (Author)
O'Reilly Media; 1 edition (September 20, 2011)

Programming Pig
Alan Gates (Author)
O'Reilly Media; 1st Edition (October, 2011)

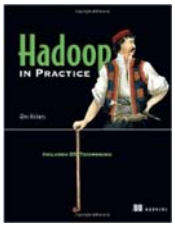


Hadoop Operations
Eric Sammer (Author)
O'Reilly Media (October 22, 2012)

Resources: Books

Data-Intensive Text Processing with MapReduce
Jimmy Lin and Chris Dyer (Authors) (April, 2010)
Download for FREE:
http://lintool.github.com/MapReduceAlgorithms/index.html

Programming Hive
Edward Capriolo, Dean Wampler,
Jason Rutherglen (Authors)
O'Reilly Media; 1 edition (October, 2012)



Hadoop in Practice
Alex Holmes (Author)
Manning Publications; (October 10, 2012)

Resources: Your Instructor

- **Dima May**
  - dimamay@coreservlets.com
  - Email me any time!

Wrap-Up

Summary

- **We learned about**
  - Data storage needs are rapidly increasing
  - Hadoop has become the de-facto standard for handling these massive data sets
  - The Cloudera Distribution for Hadoop (CDH) is the most commonly used Hadoop release distribution
  - There is a number of Hadoop related publications available

Questions?