cloudera®
Ask Bigger Questions

Cloudera Data Analyst Training: Using Pig, Hive, and Impala with Hadoop

201410

cloudera®
Ask Bigger Questions

Introduction

Chapter 1

Course Chapters

Introduction Hadoop Fundamentals	**Course Introduction**
Introduction to Pig Basic Data Analysis with Pig Processing Complex Data with Pig Multi-Dataset Operations with Pig Pig Troubleshooting and Optimization	Data ETL and Analysis With Pig
Introduction to Impala and Hive Querying With Impala and Hive Impala and Hive Data Management Data Storage and Performance	Introduction to Impala and Hive
Relational Data Analysis With Impala and Hive Working with Impala Analyzing Text and Complex Data with Hive Hive Optimization Extending Hive	Data Analysis With Impala and Hive
Choosing the Best Tool for the Job Conclusion	Course Conclusion

Chapter Topics

Introduction | **Course Introduction**

- **About This Course**
- About Cloudera
- Course Logistics
- Introductions

Course Objectives (1)

During this course, you will learn

- **The purpose of Hadoop and its related tools**
- **The features that Pig, Hive, and Impala offer for data acquisition, storage, and analysis**
- **How to identify typical use cases for large-scale data analysis**
- **How to load data from relational databases and other sources**
- **How to manage data in HDFS and export it for use with other systems**
- **How Pig, Hive, and Impala improve productivity for typical analysis tasks**
- **The language syntax and data formats supported by these tools**

Course Objectives (2)

- How to design and execute queries on data stored in HDFS
- How to join diverse datasets to gain valuable business insight
- How Hive and Impala can be extended with custom functions and scripts
- How to analyze structured, semi-structured, and unstructured data
- How to store and query data for better performance
- How to determine which tool is the best choice for a given task

Chapter Topics

Introduction | **Course Introduction**

- About This Course
- **About Cloudera**
- Course Logistics
- Introductions

About Cloudera (1)

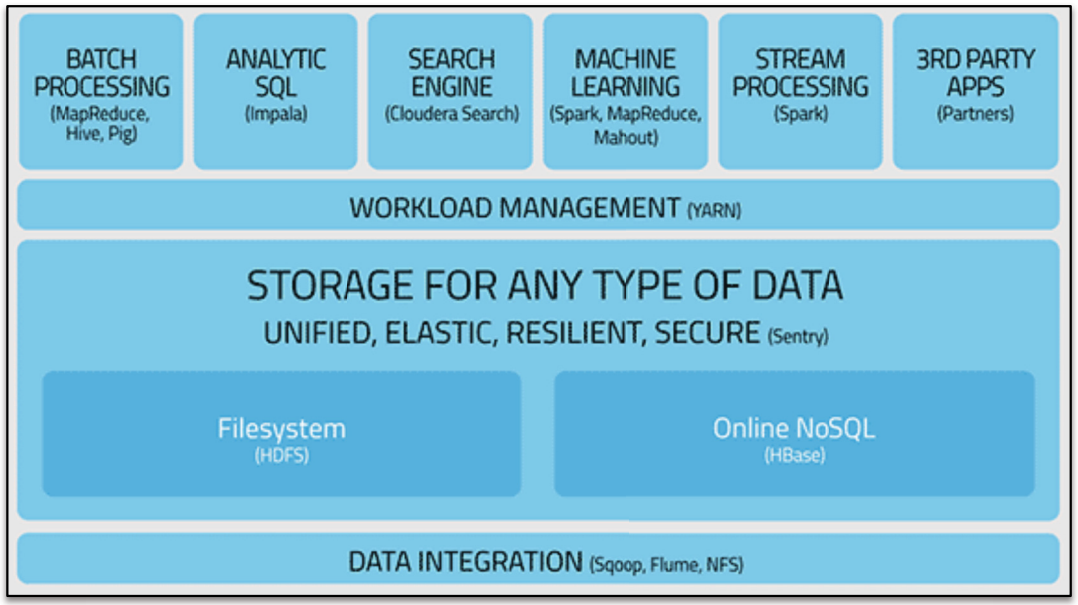
- **The leader in Apache Hadoop-based software and services**
- **Founded by leading experts on Hadoop from Facebook, Yahoo, Google, and Oracle**
- **Provides support, consulting, training, and certification for Hadoop users**
- **Staff includes committers to virtually all Hadoop projects**
- **Many authors of industry standard books on Apache Hadoop projects**
  - Tom White, Lars George, Kathleen Ting, etc.

About Cloudera (2)

- **Customers include many key users of Hadoop**
  - Allstate, AOL Advertising, Box, BT, CBS Interactive, eBay, Experian, FICO, Groupon, MasterCard, National Cancer Institute, Orbitz, Social Security Administration, Trend Micro, Trulia, US Army, …
- **Cloudera public training:**
  - Cloudera Developer Training for Apache Hadoop
  - Cloudera Developer Training for Apache Spark
  - Designing and Building Big Data Applications
  - Cloudera Administrator Training for Apache Hadoop
  - Cloudera Data Analyst Training: Using Pig, Hive, and Impala with Hadoop
  - Cloudera Training for Apache HBase
  - Introduction to Data Science: Building Recommender Systems
  - Cloudera Essentials for Apache Hadoop
- **Onsite and custom training is also available**

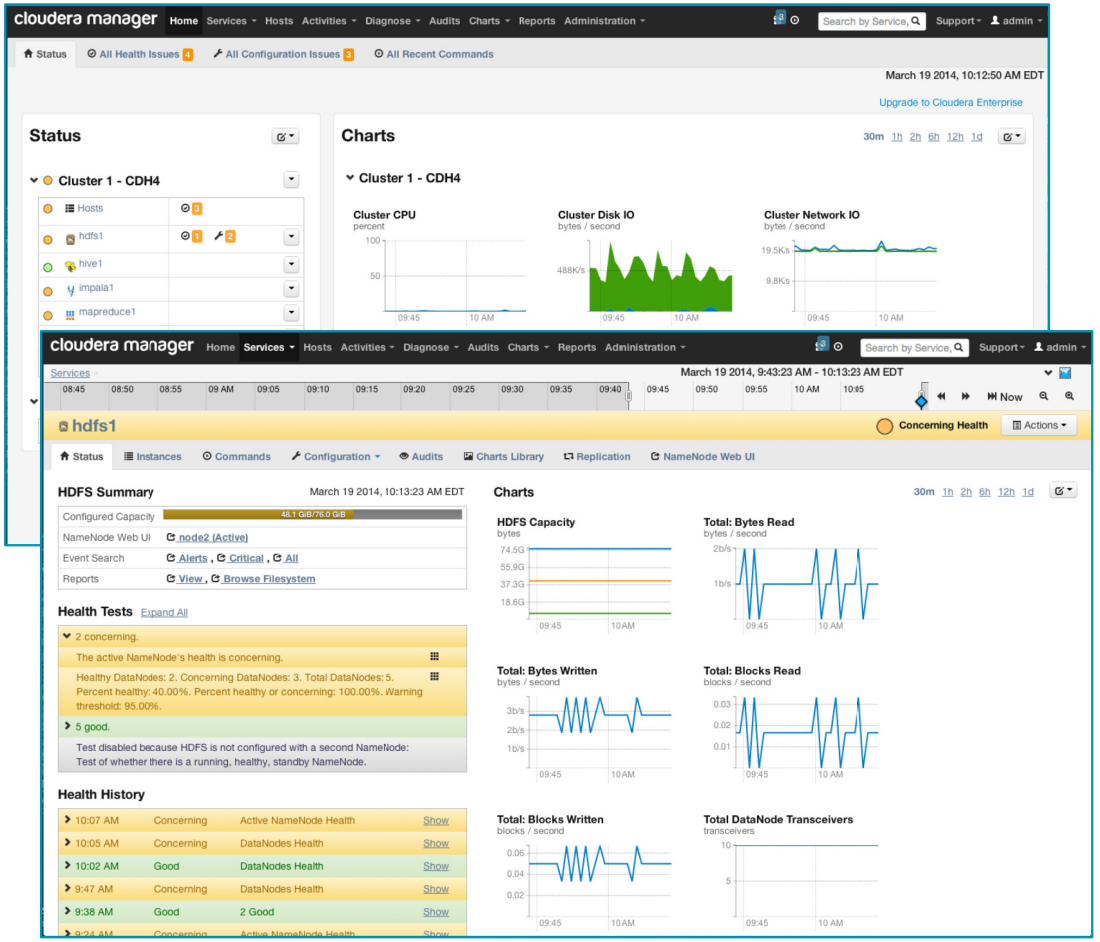
CDH

- **CDH (Cloudera's Distribution, including Apache Hadoop)**
  - 100% open source, enterprise-ready distribution of Hadoop and related projects
  - The most complete, tested, and widely-deployed distribution of Hadoop
  - Integrates all key Hadoop ecosystem projects
  - Available as RPMs and Ubuntu/Debian/SuSE packages or as a tarball



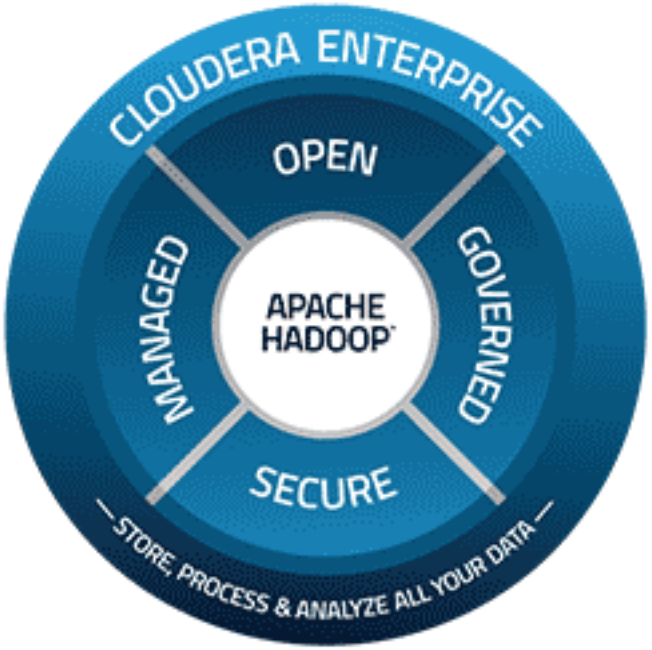
Cloudera Express

- **Cloudera Express**
  - Free download
- **The best way to get started with Hadoop**
- **Includes CDH**
- **Includes Cloudera Manager**
  - End-to-end administration for Hadoop
  - Deploy, manage, and monitor your cluster



Cloudera Enterprise

- **Cloudera Enterprise**
  - Subscription product including CDH and Cloudera Manager
- **Includes support**
- **Includes extra Cloudera Manager features**
  - Configuration history and rollbacks
  - Rolling updates
  - LDAP integration
  - SNMP support
  - Automated disaster recovery
  - Etc.



Chapter Topics

Introduction | **Course Introduction**

- About This Course
- About Cloudera
- **Course Logistics**
- Introductions

Logistics

- **Class start and finish times**
- **Lunch**
- **Breaks**
- **Restrooms**
- **Wi-Fi access**
- **Virtual machines**
- **Can I come in early/stay late?**

Your instructor will give you details on how to access the course materials and exercise instructions for the class

Chapter Topics

Introduction | **Course Introduction**

- About This Course
- About Cloudera
- Course Logistics
- **Introductions**

Introductions

- **About your instructor**
- **About you**
  - Where do you work and what do you do there?
  - Which database(s) and platform(s) do you use?
  - Have you worked with Apache Hadoop or related tools?
  - Any experience as a developer?
    - What programming languages do you use?
  - What are your expectations for this course?

cloudera®
Ask Bigger Questions

Hadoop Fundamentals

Chapter 2

Course Chapters

▪ Introduction ▪ **Hadoop Fundamentals**	**Course Introduction**
▪ Introduction to Pig ▪ Basic Data Analysis with Pig ▪ Processing Complex Data with Pig ▪ Multi-Dataset Operations with Pig ▪ Pig Troubleshooting and Optimization	Data ETL and Analysis With Pig
▪ Introduction to Impala and Hive ▪ Querying With Impala and Hive ▪ Impala and Hive Data Management ▪ Data Storage and Performance	Introduction to Impala and Hive
▪ Relational Data Analysis With Impala and Hive ▪ Working with Impala ▪ Analyzing Text and Complex Data with Hive ▪ Hive Optimization ▪ Extending Hive	Data Analysis With Impala and Hive
▪ Choosing the Best Tool for the Job ▪ Conclusion	Course Conclusion

Hadoop Fundamentals

In this chapter, you will learn

- **Which factors led to the era of Big Data**
- **What Hadoop is and what significant features it offers**
- **How Hadoop offers reliable storage for massive amounts of data with HDFS**
- **How Hadoop supports large-scale data processing through MapReduce**
- **How 'Hadoop Ecosystem' tools can boost an analyst's productivity**
- **Several ways to integrate Hadoop into the modern data center**

Chapter Topics

Velocity

- **We are generating data faster than ever**
  - Processes are increasingly automated
  - Systems are increasingly interconnected
  - People are increasingly interacting online

Variety

- **We are producing a wide variety of data**
  - Social network connections
  - Server and application log files
  - Electronic medical records
  - Images, audio, and video
  - RFID and wireless sensor network events
  - Product ratings on shopping and review Web sites
  - And much more…
- **Not all of this maps cleanly to the relational model**

Volume

- **Every day…**
  - More than 1.5 billion shares are traded on the New York Stock Exchange
  - Facebook stores 2.7 billion comments and 'Likes'
  - Google processes about 24 petabytes of data
- **Every minute…**
  - Foursquare handles more than 2,000 check-ins
  - TransUnion makes nearly 70,000 updates to credit files
- **And every second…**
  - Banks process more than 10,000 credit card transactions

Data Has Value

- **This data has many valuable applications**
  - Product recommendations
  - Predicting demand
  - Marketing analysis
  - Fraud detection
  - And many, many more…
- **We must process it to extract that value**
  - And processing *all the data* can yield more accurate results

We Need a System that Scales

- **We're generating too much data to process with traditional tools**
- **Two key problems to address**
  - How can we reliably store large amounts of data at a reasonable cost?
  - How can we analyze all the data we have stored?

Chapter Topics

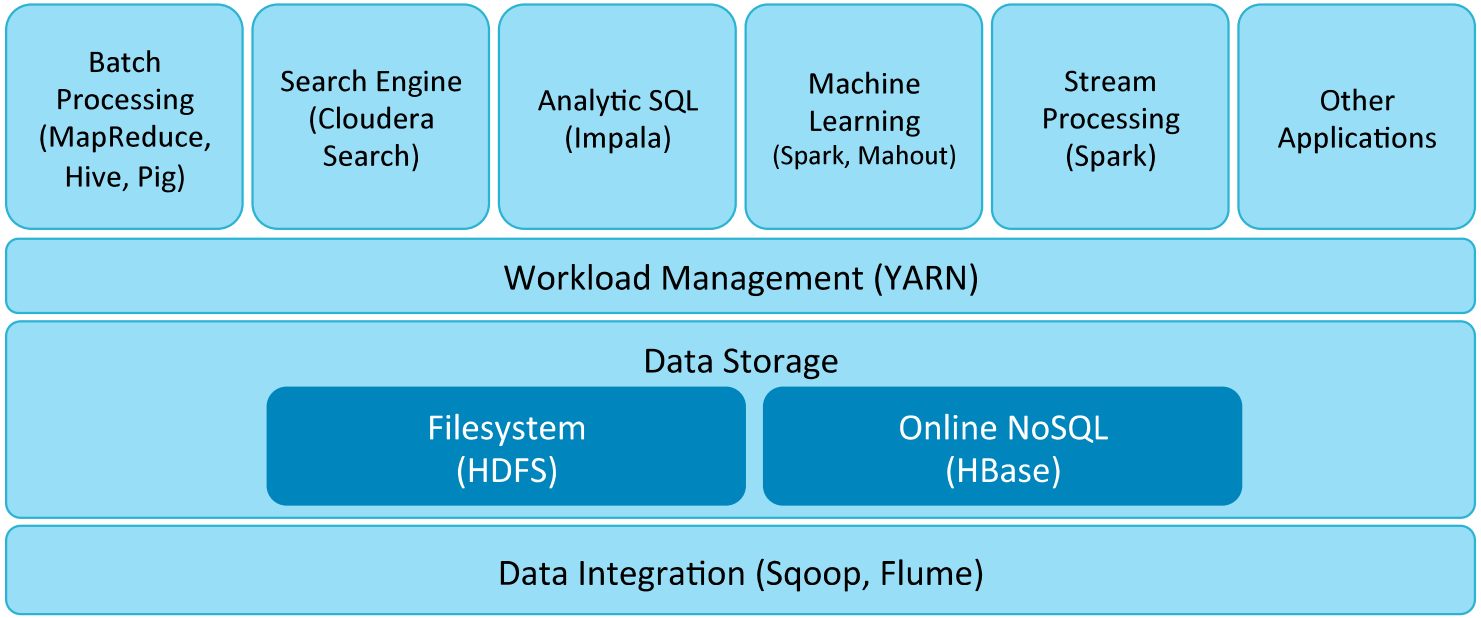
Hadoop Fundamentals | **Course Introduction**

- The Motivation for Hadoop
- **Hadoop Overview**
- Data Storage: HDFS
- Distributed Data Processing: YARN, MapReduce, and Spark
- Data Processing and Analysis: Pig, Hive, and Impala
- Database Integration: Sqoop
- Other Hadoop Data Tools
- Exercise Scenario Explanation
- Conclusion
- Hands-On Exercise: Data Ingest with Hadoop Tools

What is Apache Hadoop?

- **Scalable and economical data storage and processing**
  - Distributed and fault-tolerant
  - Harnesses the power of industry standard hardware

- **Heavily inspired by technical documents published by Google**



Scalability

- **Hadoop is a distributed system**
  - A collection of servers running Hadoop software is called a *cluster*
- **Individual servers within a cluster are called *nodes***
  - Typically standard rackmount servers running Linux
  - Each node both stores and processes data
- **Add more nodes to the cluster to increase scalability**
  - A cluster may contain up to several thousand nodes
  - You can scale out incrementally as required

Fault Tolerance

- **Paradox: Adding nodes increases the chance that any one of them will fail**
  - Solution: build redundancy into the system and handle it automatically
- **Files loaded into HDFS are replicated across nodes in the cluster**
  - If a node fails, its data is re-replicated using one of the other copies
- **Data processing jobs are broken into individual tasks**
  - Each task takes a small amount of data as input
  - Thousands of tasks (or more) often run in parallel
  - If a node fails during processing, its tasks are rescheduled elsewhere
- **Routine failures are handled automatically without any loss of data**

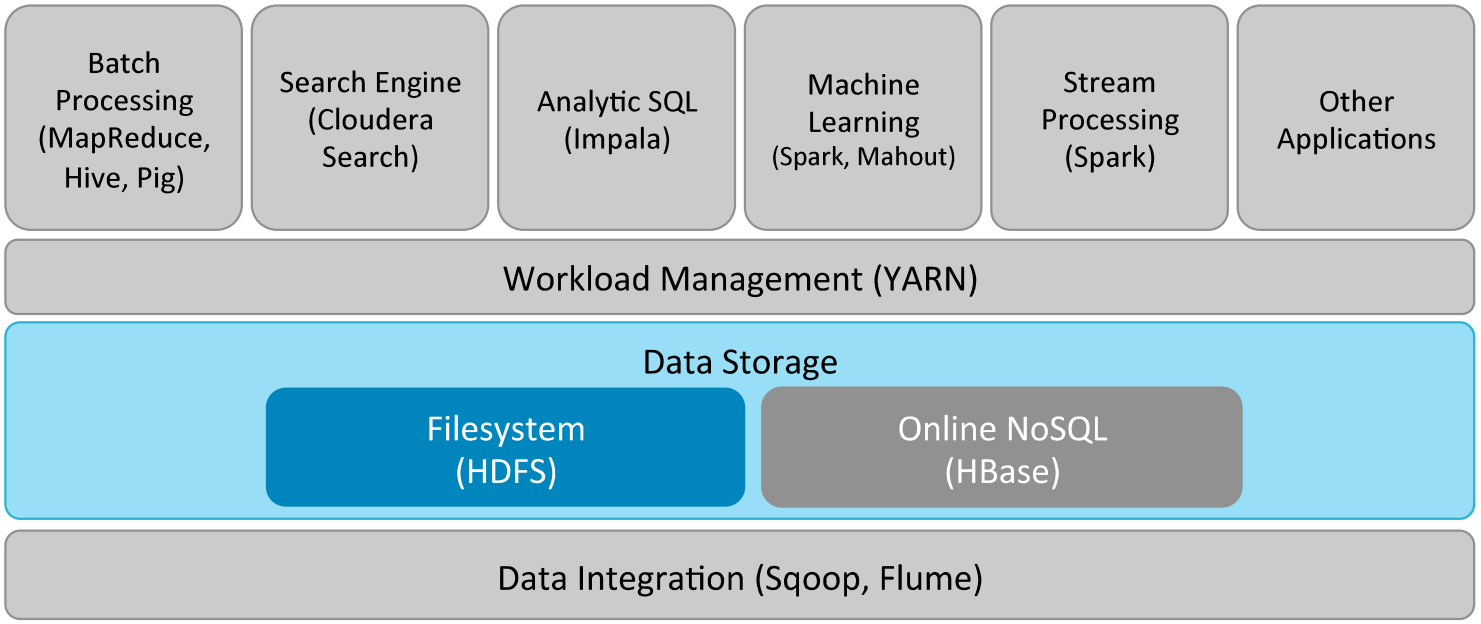
Chapter Topics

Hadoop Fundamentals | **Course Introduction**

- The Motivation for Hadoop
- Hadoop Overview
- **Data Storage: HDFS**
- Distributed Data Processing: YARN, MapReduce, and Spark
- Data Processing and Analysis: Pig, Hive, and Impala
- Database Integration: Sqoop
- Other Hadoop Data Tools
- Exercise Scenario Explanation
- Conclusion
- Hands-On Exercise: Data Ingest with Hadoop Tools

HDFS: Hadoop Distributed File System

- **HDFS provides the storage layer for Hadoop data processing**
- **Provides inexpensive and reliable storage for massive amounts of data**
- **Other Hadoop components work with data in HDFS**
  - MapReduce, Impala, Hive, Pig, Spark, etc.



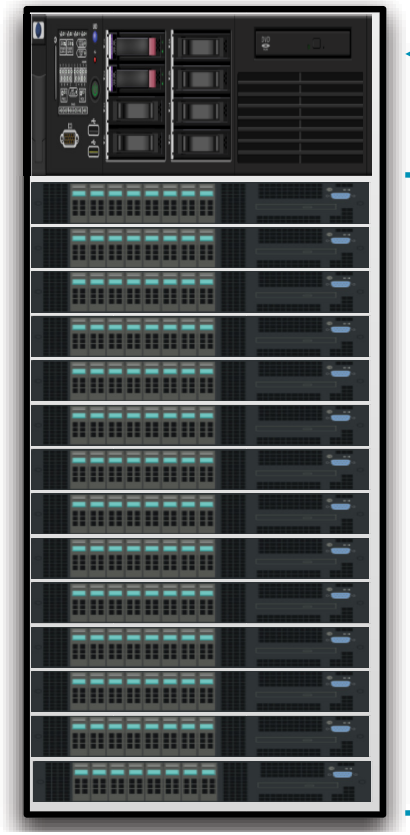
HDFS Features

- **Optimized for sequential access to a relatively small number of large files**
  - Each file is likely to be 100MB or larger
  - Multi-gigabyte files are typical
- **In some ways, HDFS is similar to a UNIX filesystem**
  - Hierarchical, with UNIX-style paths (e.g., `/sales/rpt/asia.txt`)
  - UNIX-style file ownership and permissions
- **There are also some major deviations from UNIX**
  - No concept of a current directory
  - Cannot modify files once written
  - Must use Hadoop-specific utilities or custom code to access HDFS

HDFS Architecture

- **Hadoop has a master/slave architecture**
- **HDFS master daemon: NameNode**
  - Manages namespace and *metadata*
  - Monitors slave nodes
- **HDFS slave daemon: DataNode**
  - Reads and writes the actual data

A Small Hadoop Cluster



Master
HDFS master daemon

Slaves
HDFS slave daemons

Accessing HDFS via the Command Line

- **HDFS is not a general purpose filesystem**
  - Not built into the OS, so only specialized tools can access it
  - End users typically access HDFS via the `hdfs dfs` command
- **Example: display the contents of the `/user/fred/sales.txt` file**

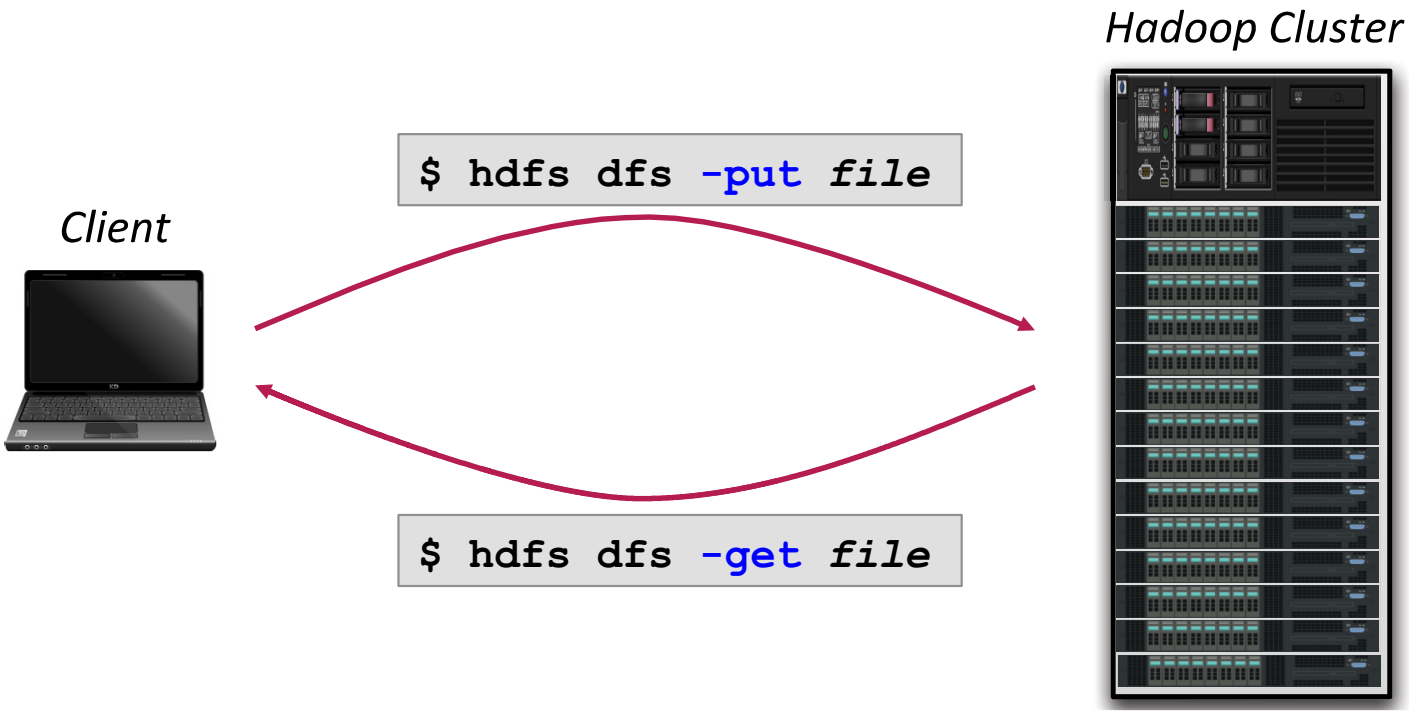
```
$ hdfs dfs -cat /user/fred/sales.txt
```

- **Example: Create a directory (below the root) called `reports`**

```
$ hdfs dfs -mkdir /reports
```

Copying Local Data To and From HDFS

- **Remember that HDFS is distinct from your local filesystem**
  - Use **`hdfs dfs -put`** to copy local files **to** HDFS
  - Use **`hdfs dfs -get`** to fetch a local copy of a file **from** HDFS



More `hdfs dfs` Command Examples

- **Copy file `input.txt` from local disk to the user's directory in HDFS**

```
$ hdfs dfs -put input.txt input.txt
```

  - This will copy the file to `/user/username/input.txt`

- **Get a directory listing of the HDFS root directory**

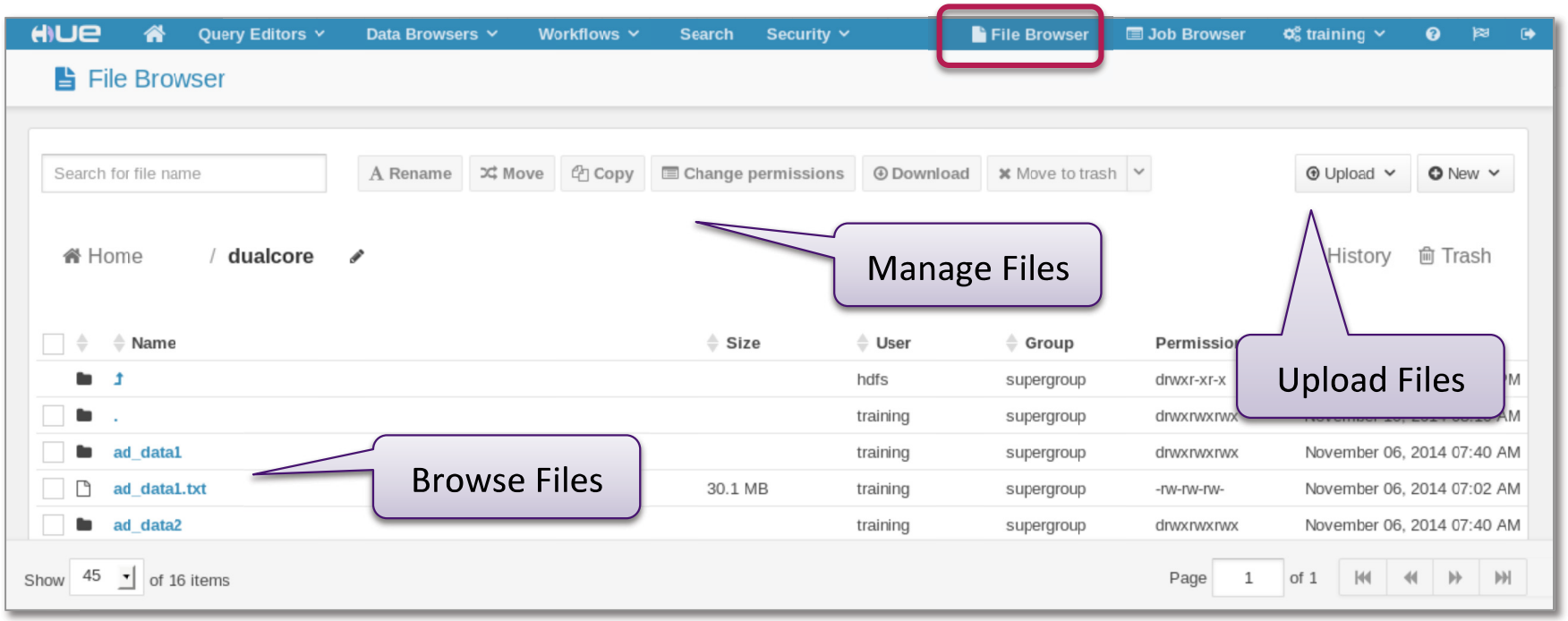
```
$ hdfs dfs -ls /
```

- **Delete the file `/reports/sales.txt`**

```
$ hdfs dfs -rm /reports/sales.txt
```

Using the Hue HDFS File Manager

- **Hue is a Web interface for Hadoop**
  - **H**adoop **U**ser **E**xperience
- **Hue includes an application for browsing and managing files in HDFS**
  - To use Hue, browse to `http://hue_server:8888/`



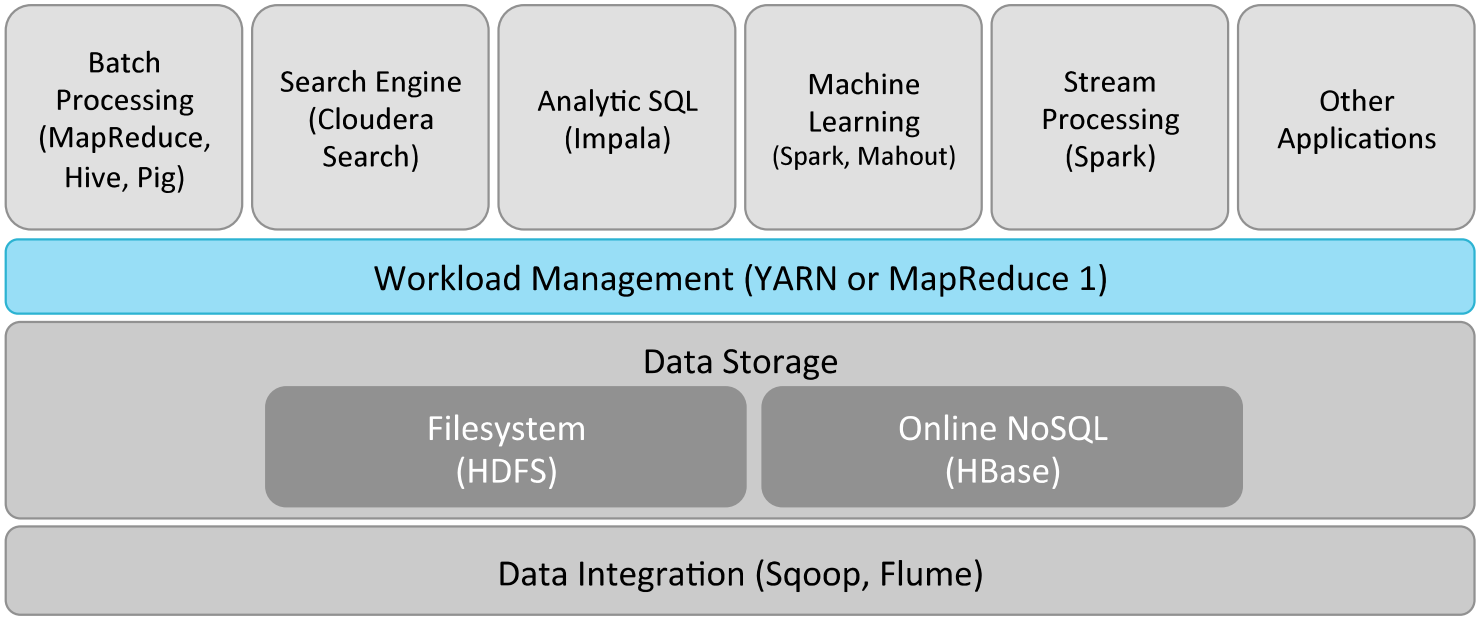
Chapter Topics

Hadoop Fundamentals | **Course Introduction**

- The Motivation for Hadoop
- Hadoop Overview
- Data Storage: HDFS
- **Distributed Data Processing: YARN, MapReduce, and Spark**
- Data Processing and Analysis: Pig, Hive, and Impala
- Database Integration: Sqoop
- Other Hadoop Data Tools
- Exercise Scenario Explanation
- Conclusion
- Hands-On Exercise: Data Ingest with Hadoop Tools

Workload Management: YARN

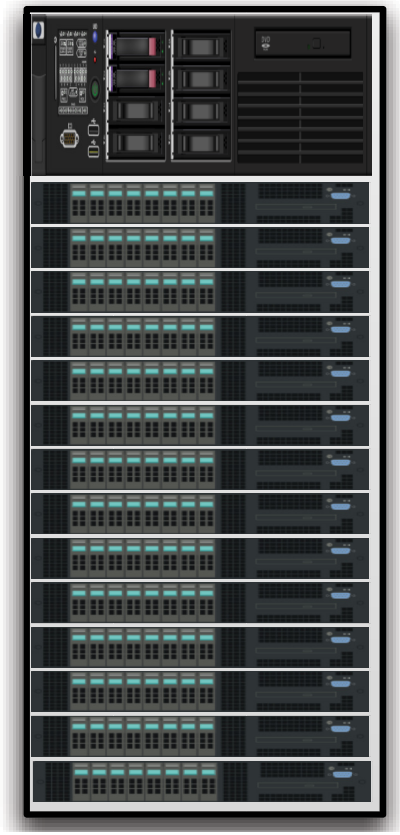
- **Many Hadoop tools work with data in a Hadoop cluster**
- **Requires workload management to distribute and monitor work across the cluster**



Hadoop Cluster Architecture

- **Master/Slave Architecture**
  - YARN or MapReduce version 1
    - Details differ slightly
- **Master nodes**
  - Run master daemons to accept jobs, and monitor and distribute work
- **Slave nodes**
  - Run slave daemons to start tasks
  - Do the actual work
  - Report status back to master daemons
- **HDFS and YARN/MRv1 are collocated**
  - Slave nodes run both HDFS and slave daemons on the same machines

A Small Hadoop Cluster

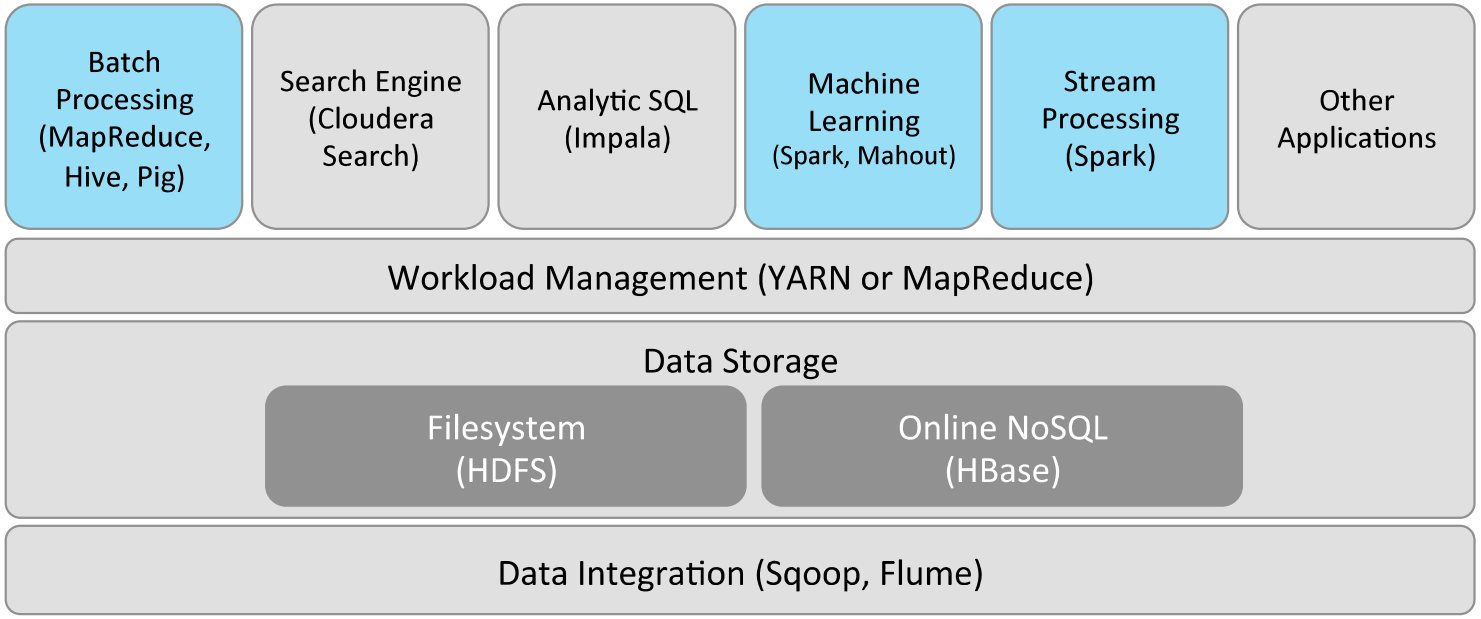


Master
YARN master daemon
HDFS master daemon

Slaves
YARN slave daemons
HDFS slave daemons

General Data Processing

- **Hadoop includes two general data processing engines**
  - MapReduce
  - Spark
- **Both are programming libraries (Java, Scala, Python…)**



Hadoop MapReduce

- **Hadoop MapReduce was the original processing engine for Hadoop**
  - Still the most commonly used general data processing engine
- **Based on the the 'map-reduce' programming model**
  - A style of processing data popularized by Google
- **Provides a set of programming libraries**
  - Primarily supports Java
  - Streaming MapReduce provides (limited) support for scripting languages such as Python
- **Benefits of Hadoop MapReduce**
  - Simplicity
  - Flexibility
  - Scalability

Apache Spark

- **The next generation general data processing engine**
- **Builds on the same 'map-reduce' programming model as Hadoop MapReduce**
- **Originally developed at AMP Lab at UC Berkeley**
- **Spark supports Scala, Java, and Python**
- **Spark has the same benefits as MapReduce, plus…**
  - Improved performance using in-memory processing
  - Higher level programming model to speed up development

Chapter Topics

Hadoop Fundamentals | **Course Introduction**

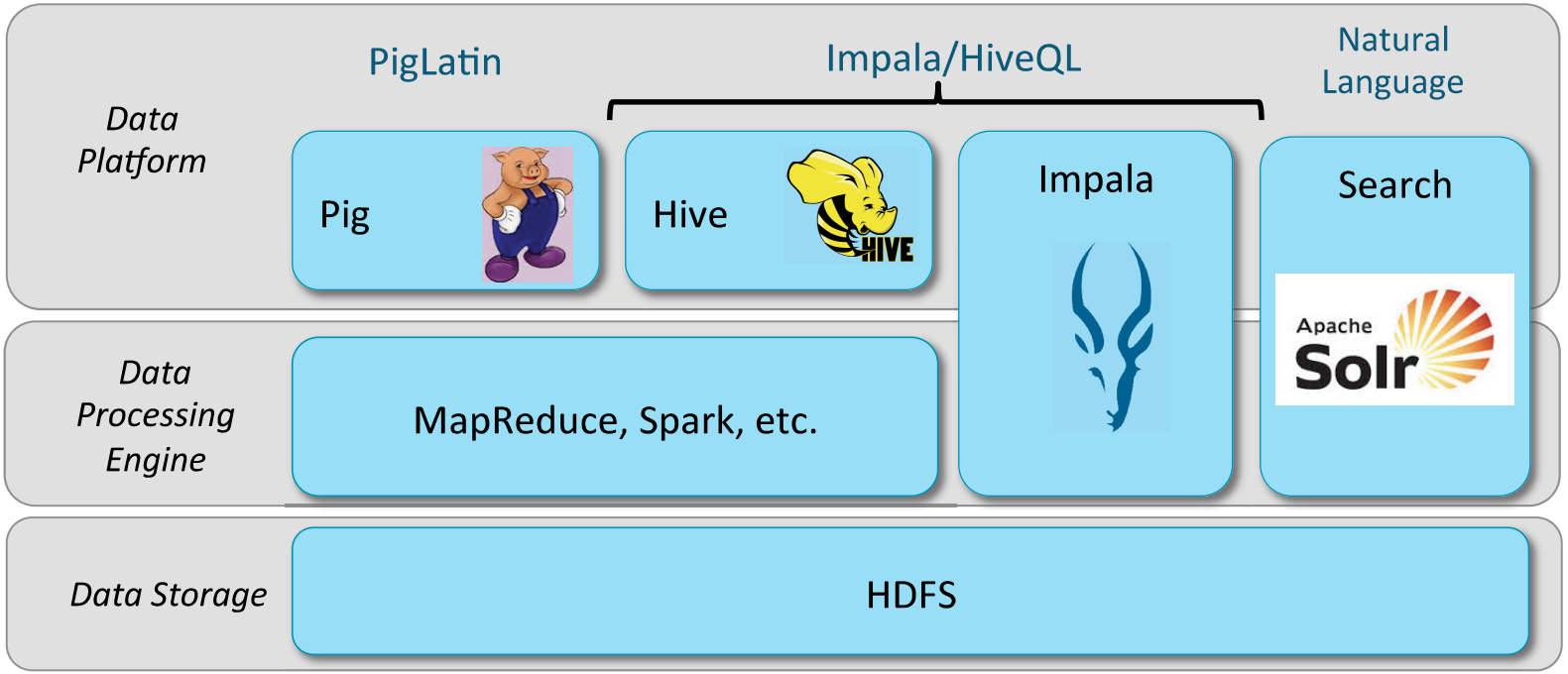
- The Motivation for Hadoop
- Hadoop Overview
- Data Storage: HDFS
- Distributed Data Processing: YARN, MapReduce, and Spark
- **Data Analysis and Processing: Pig, Hive, and Impala**
- Database Integration: Sqoop
- Other Hadoop Data Tools
- Exercise Scenario Explanation
- Conclusion
- Hands-On Exercise: Data Ingest with Hadoop Tools

Data Processing and Analysis with Hadoop (1)

- **Hadoop MapReduce and Spark are powerful data processing engines but…**
  - Hard to master
  - Require programming skills
  - Slow to develop, hard to maintain
- **Hadoop includes several other tools for data processing and analysis**
  - Tools for data analysts, not programmers

Data Processing and Analysis with Hadoop (2)

- **Higher level abstractions for general data processing**
  - Pig, Hive
- **Specialized processing engines for interactive analysis**
  - Impala, Search



Apache Pig

- **Apache Pig builds on Hadoop to offer high-level data processing**
  - This is an alternative to writing low-level MapReduce code
  - Pig is especially good at joining and transforming data

```
people = LOAD '/user/training/customers' AS (cust_id, name);
orders = LOAD '/user/training/orders' AS (ord_id, cust_id, cost);
groups = GROUP orders BY cust_id;
totals = FOREACH groups GENERATE group, SUM(orders.cost) AS t;
result = JOIN totals BY group, people BY cust_id;
DUMP result;
```

- **The Pig interpreter runs on the client machine**
  - Turns PigLatin scripts into MapReduce jobs
  - Submits those jobs to the cluster

Apache Hive

- **Hive is another abstraction on top of Hadoop**
  - Like Pig, it also reduces development time
  - Hive uses a SQL-like language called HiveQL

```
  SELECT customers.cust_id, SUM(cost) AS total
    FROM customers
    JOIN orders
      ON (customers.cust_id = orders.cust_id)
GROUP BY customers.cust_id
ORDER BY total DESC;
```

- **A Hive Server runs on a master node**
  - Turns HiveQL queries into MapReduce jobs
  - Submits those jobs to the cluster

Cloudera Impala

- **Massively parallel SQL engine which runs on a Hadoop cluster**
  - Inspired by Google's Dremel project
  - Can query data stored in HDFS or HBase tables
- **Uses Impala SQL**
  - Very similar to HiveQL
- **High performance**
  - Typically at least 10 times faster than Hive or MapReduce
  - High-level query language (subset of SQL-92)
- **Impala is 100% Apache-licensed open source**

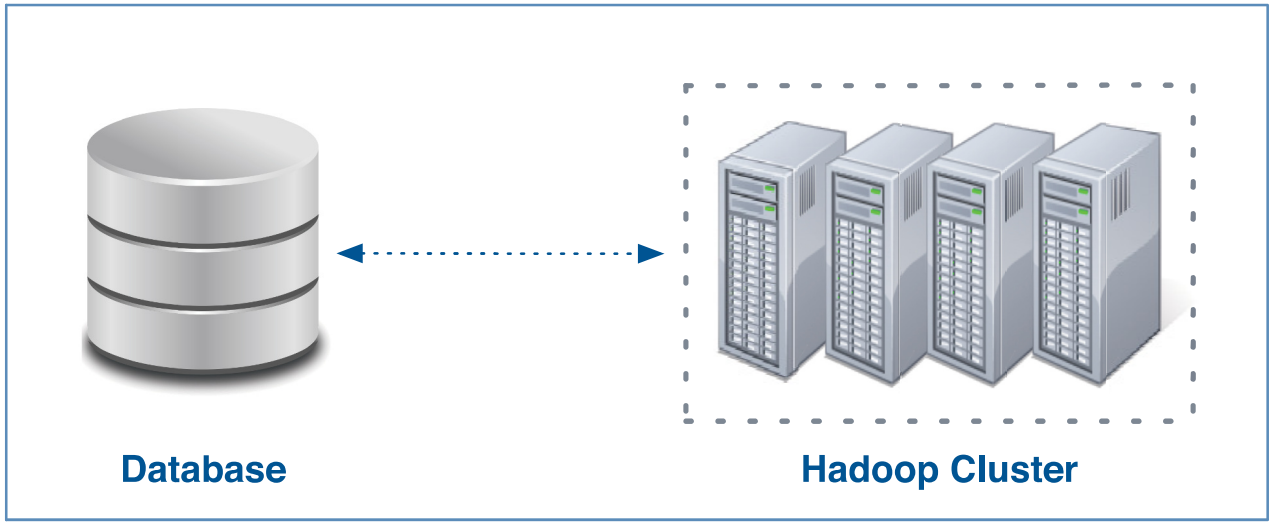
Chapter Topics

Hadoop Fundamentals | **Course Introduction**

- The Motivation for Hadoop
- Hadoop Overview
- Data Storage: HDFS
- Distributed Data Processing: YARN, MapReduce, and Spark
- Data Processing and Analysis: Pig, Hive, and Impala
- **Database Integration: Sqoop**
- Other Hadoop Data Tools
- Exercise Scenario Explanation
- Conclusion
- Hands-On Exercise: Data Ingest with Hadoop Tools

Apache Sqoop

- **Sqoop exchanges data between an RDBMS and Hadoop**
- **It can import all tables, a single table, or a portion of a table into HDFS**
  - Does this very efficiently via a Map-only MapReduce job
  - Result is a directory in HDFS containing comma-delimited text files
- **Sqoop can also export data from HDFS back to the database**



Importing Tables with Sqoop

- **This example imports the `customers` table from a MySQL database**
  - Will create `/mydata/customers` directory in HDFS
  - Directory will contain comma-delimited text files

```
$ sqoop import \
    --connect jdbc:mysql://localhost/company \
    --username twheeler --password bigsecret \
    --warehouse-dir /mydata \
    --table customers
```

- **Adding the `--direct` option may offer better performance**
  - Uses database-specific tools instead of JDBC
  - This option is not compatible with all databases
- **High-performance custom connectors for some databases**
  - Netezza, Teradata, MySQL…

Importing An Entire Database with Sqoop

- **Import all tables from the database (fields will be tab-delimited)**

```
$ sqoop import-all-tables \
    --connect jdbc:mysql://localhost/company \
    --username twheeler --password bigsecret \
    --fields-terminated-by '\t' \
    --warehouse-dir /mydata
```

Importing Partial Tables with Sqoop

- **Import only specified columns from `products` table**

```
$ sqoop import \
    --connect jdbc:mysql://localhost/company \
    --username twheeler --password bigsecret \
    --warehouse-dir /mydata \
    --table products \
    --columns "prod_id,name,price"
```

- **Import only matching rows from `products` table**

```
$ sqoop import \
    --connect jdbc:mysql://localhost/company \
    --username twheeler --password bigsecret \
    --warehouse-dir /mydata \
    --table products \
    --where "price >= 1000"
```

Incremental Imports with Sqoop

- **What if new records are added to the database?**
  - Could re-import all records, but this is inefficient
- **Sqoop's incremental append mode imports only *new* records**
  - Based on value of last record in specified column

```
$ sqoop import \
    --connect jdbc:mysql://localhost/company \
    --username twheeler --password bigsecret \
    --warehouse-dir /mydata \
    --table orders \
    --incremental append \
    --check-column order_id \
    --last-value 6713821
```

Handling Modifications with Incremental Imports

- **What if existing records are also modified in the database?**
  - Incremental append mode doesn't handle this
- **In CDH 5.2 and later, Sqoop's `lastmodified` append mode adds *and* updates records**
  - Caveat: You must maintain a timestamp column in your table

```
$ sqoop import \
    --connect jdbc:mysql://localhost/company \
    --username twheeler --password bigsecret \
    --warehouse-dir /mydata \
    --table shipments \
    --incremental lastmodified \
    --check-column last_update_date \
    --last-value "2013-06-12 03:15:59"
```

Exporting Data from Hadoop to RDBMS with Sqoop

- **We have seen several ways to pull records from an RDBMS into Hadoop**
  - It is sometimes also helpful to *push* data in Hadoop back to an RDBMS
- **Sqoop supports this via `export`**

```
$ sqoop export \
    --connect jdbc:mysql://localhost/company \
    --username twheeler --password bigsecret \
    --export-dir /mydata/recommender_output \
    --table product_recommendations
```

Chapter Topics

Hadoop Fundamentals | **Course Introduction**

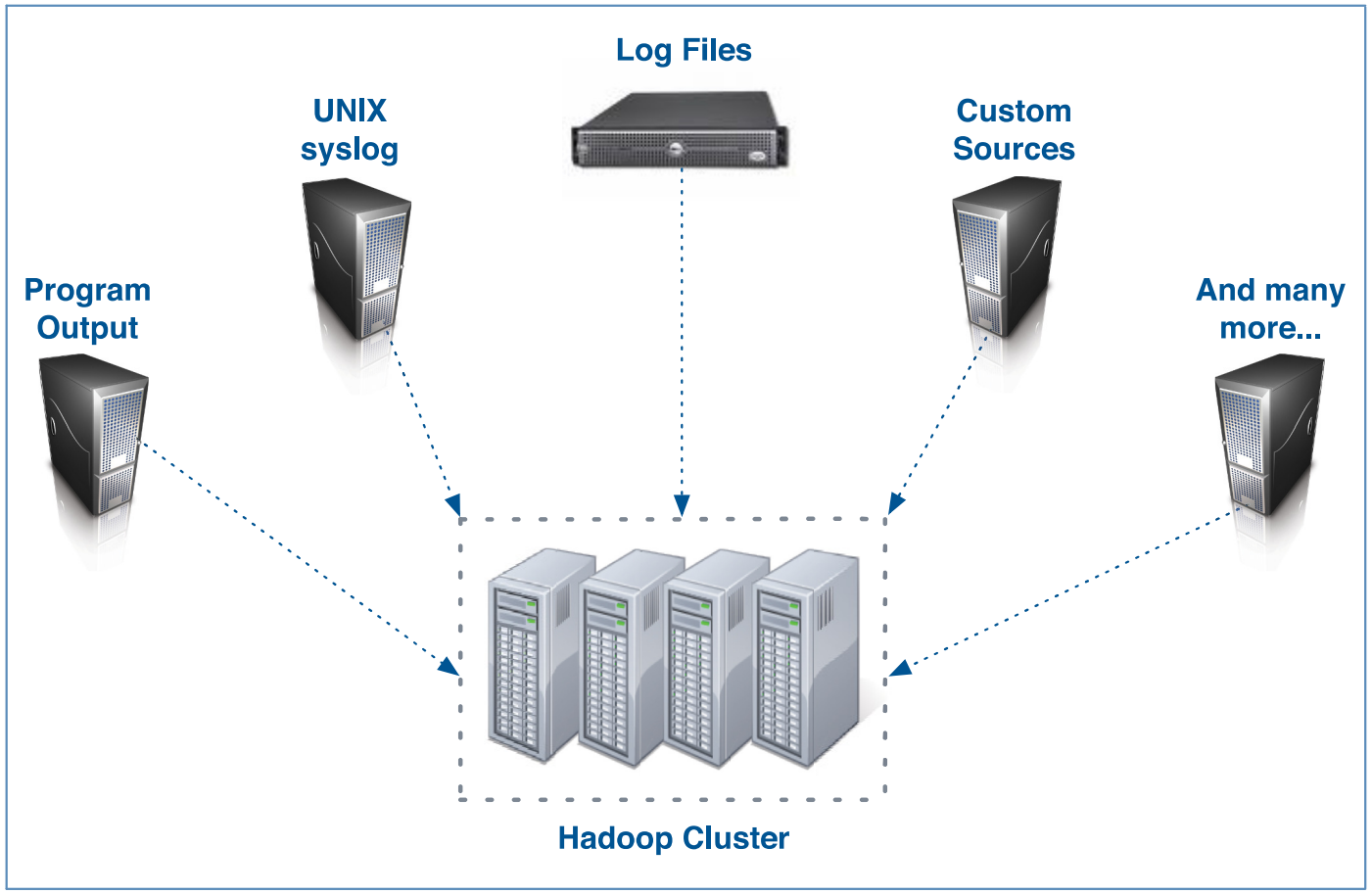
- The Motivation for Hadoop
- Hadoop Overview
- Data Storage: HDFS
- Distributed Data Processing: YARN, MapReduce, and Spark
- Data Processing and Analysis: Pig, Hive, and Impala
- Database Integration: Sqoop
- **Other Hadoop Data Tools**
- Exercise Scenario Explanation
- Conclusion
- Hands-On Exercise: Data Ingest with Hadoop Tools

Apache HBase

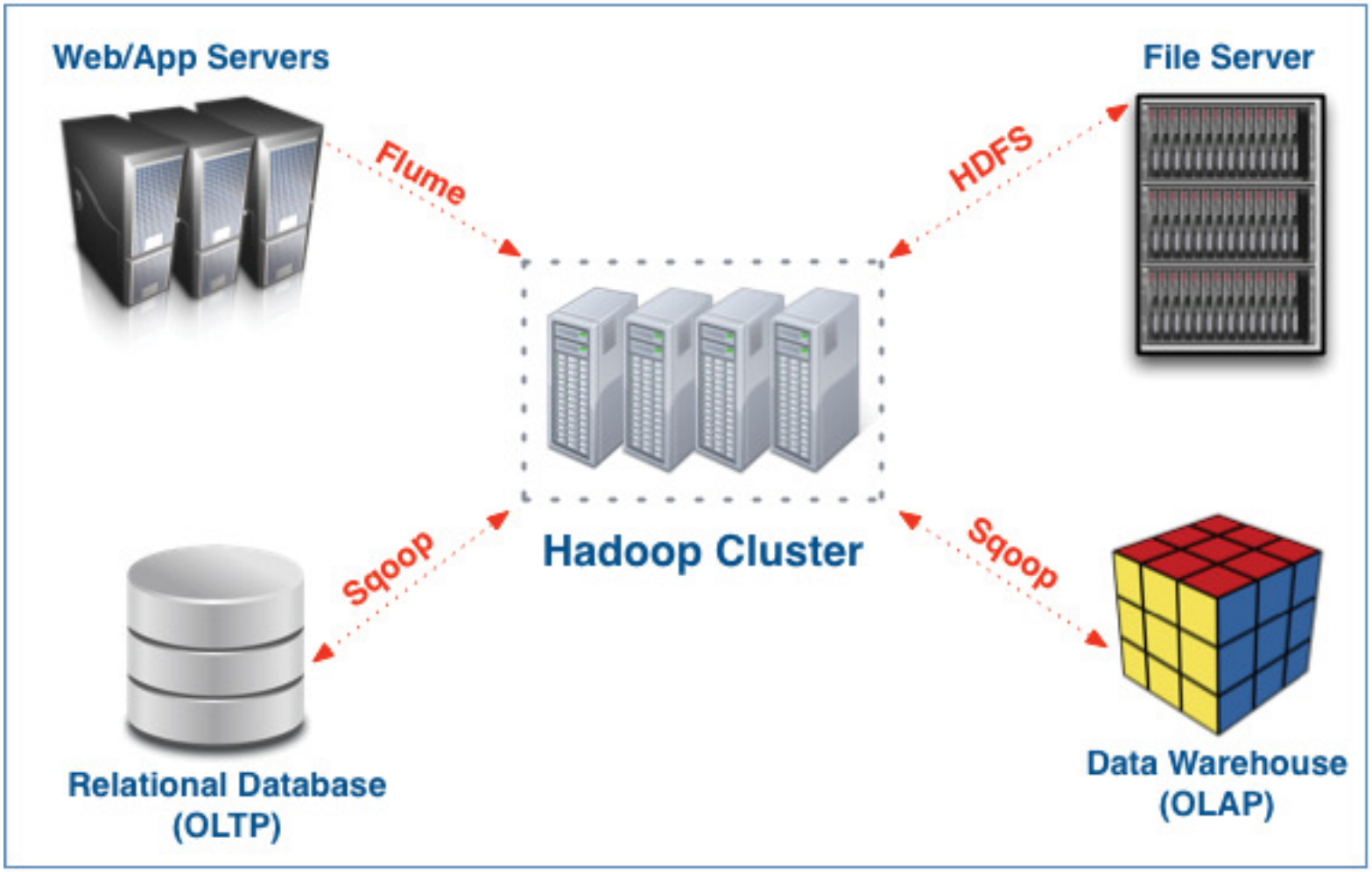
- **HBase is "the Hadoop database"**
- **Can store massive amounts of data**
  - Gigabytes, terabytes, and even petabytes of data in a table
  - Tables can have many thousands of columns
- **Scales to provide very high write throughput**
  - Hundreds of thousands of inserts per second
- **Fairly primitive when compared to an RDBMS**
  - NoSQL : There is no high-level query language
  - Use API to `scan` / `get` / `put` values based on keys

Apache Flume

- **Flume imports data into HDFS *as it is being generated* by various sources**



Recap: Data Center Integration



Chapter Topics

Hadoop Fundamentals | **Course Introduction**

- The Motivation for Hadoop
- Hadoop Overview
- Data Storage: HDFS
- Distributed Data Processing: YARN, MapReduce, and Spark
- Data Processing and Analysis: Pig, Hive, and Impala
- Database Integration: Sqoop
- Other Hadoop Data Tools
- **Exercise Scenario Explanation**
- Conclusion
- Hands-On Exercise: Data Ingest with Hadoop Tools

Hands-On Exercises: Scenario Explanation

- **Hands-On Exercises throughout the course will reinforce the topics being discussed**
  - Exercises simulate the kind of tasks often performed using the tools you will learn about in class
  - Most exercises depend on data generated in earlier exercises
- **Scenario: Dualcore Inc. is a leading electronics retailer**
  - More than 1,000 brick-and-mortar stores
  - Dualcore also has a thriving e-commerce Web site
- **Dualcore has hired you to help find value in its data**
  - You will process and analyze data from internal and external sources
  - Identify opportunities to increase revenue
  - Find new ways to reduce costs
  - Help other departments achieve their goals

Chapter Topics

Hadoop Fundamentals | **Course Introduction**

- The Motivation for Hadoop
- Hadoop Overview
- Data Storage: HDFS
- Distributed Data Processing: YARN, MapReduce, and Spark
- Data Processing and Analysis: Pig, Hive, and Impala
- Database Integration: Sqoop
- Other Hadoop Data Tools
- Exercise Scenario Explanation
- **Conclusion**
- Hands-On Exercise: Data Ingest with Hadoop Tools

Essential Points

- **We are generating more data – and faster – than ever before**
- **Most of this data maps poorly to structured relational tables**
- **The ability to store and process this data can yield valuable insight**
- **Hadoop offers scalable data storage and processing**
- **There are lots of tools in the Hadoop ecosystem that help you to integrate Hadoop with other systems, manage complex jobs, and ease analysis**

Bibliography

The following offer more information on topics discussed in this chapter

- **10 Hadoopable Problems (recorded presentation)**
  - http://tiny.cloudera.com/dac02a
- **Guide to HDFS Commands**
  - http://tiny.cloudera.com/hdfscommands
- ***Hadoop: The Definitive Guide, 3rd Edition* (O'Reilly book)**
  - http://tiny.cloudera.com/hadooptdg
- **Sqoop User Guide**
  - http://tiny.cloudera.com/sqoopuser
- **Spark Documentation**
  - http://tiny.cloudera.com/sparkdoc

Chapter Topics

Hadoop Fundamentals | **Course Introduction**

- The Motivation for Hadoop
- Hadoop Overview
- Data Storage: HDFS
- Distributed Data Processing: YARN, MapReduce, and Spark
- Data Processing and Analysis: Pig, Hive, and Impala
- Database Integration: Sqoop
- Other Hadoop Data Tools
- Exercise Scenario Explanation
- Conclusion
- **Hands-On Exercise: Data Ingest with Hadoop Tools**

About the Training Virtual Machine

- **During this course, you will perform numerous hands-on exercises using the Cloudera Training Virtual Machine (VM)**
- **The VM has Hadoop installed in *pseudo-distributed mode***
  - A cluster comprised of a single node
  - Typically used for testing code before deploying to a large cluster

Hands-On Exercise: Data Ingest with Hadoop Tools

- **In this Hands-On Exercise, you will gain practice adding data from the local filesystem and a relational database server to HDFS**
  - You will analyze this data in subsequent exercises
- **Please refer to the Hands-On Exercise Manual for instructions**

cloudera®
Ask Bigger Questions

Introduction to Pig

Chapter 3

Course Chapters

▪ Introduction ▪ Hadoop Fundamentals	Course Introduction
▪ **Introduction to Pig** ▪ Basic Data Analysis with Pig ▪ Processing Complex Data with Pig ▪ Multi-Dataset Operations with Pig ▪ Pig Troubleshooting and Optimization	**Data ETL and Analysis With Pig**
▪ Introduction to Impala and Hive ▪ Querying With Impala and Hive ▪ Impala and Hive Data Management ▪ Data Storage and Performance	Introduction to Impala and Hive
▪ Relational Data Analysis With Impala and Hive ▪ Working with Impala ▪ Analyzing Text and Complex Data with Hive ▪ Hive Optimization ▪ Extending Hive	Data Analysis With Impala and Hive
▪ Choosing the Best Tool for the Job ▪ Conclusion	Course Conclusion