Overview

Prof. Hyuk-Yoon Kwon
https://sites.google.com/view/seoultech-bigdata

Sources: Cloudera

Contents

■ Part1: Course Overview

- Database Practice: NoSQL, Hadoop, Big data management systems
- Web Programming: PHP with Database

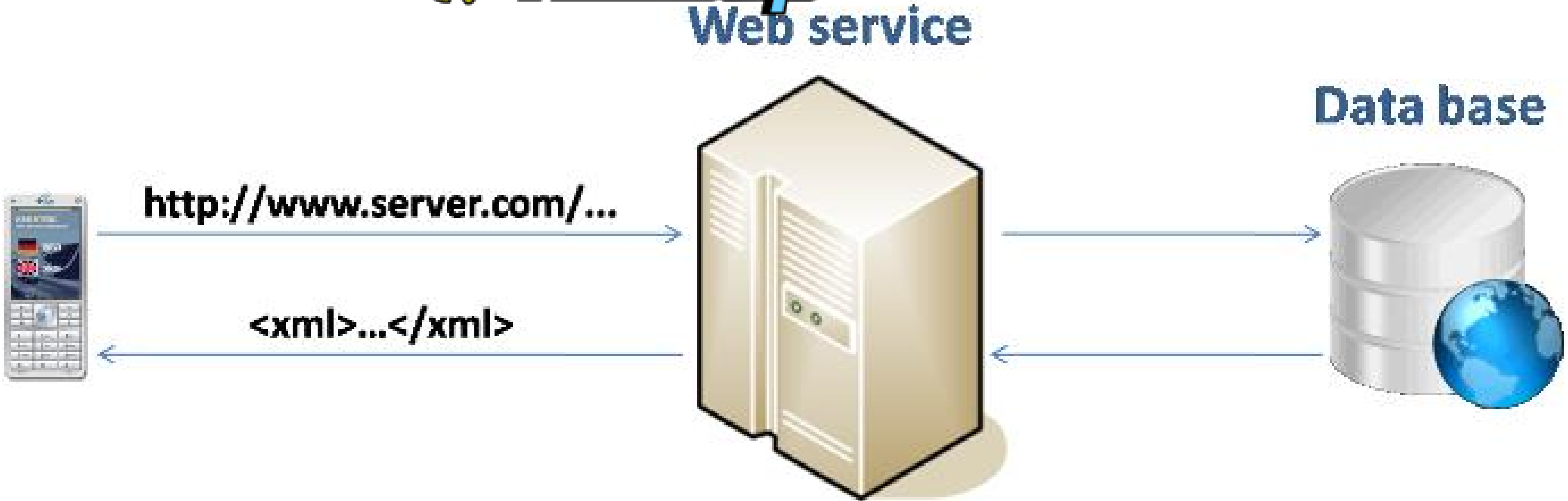
■ Part2: Introduction to Databases

Course Theme

- To learn and practice big software development skills focusing on big data management
- To learn how to deal with the big data management systems and how to build Web services based on the database systems

cloudera

hadoop



The Aim of Course

1. **To learn big data management systems and practice them using real data sets**

2. **To make real Web services based on the database and big data management systems**

Course Policy: HW and Projects

- **Penalty for cheating**
  - The lowest grade
- **What is cheating?**
  - Sharing code: by copying, retyping, looking at, or supplying a file
  - Searching the Web for solutions
  - Helping your friend to write a lab, line by line
- **What is NOT cheating?**
  - Explaining how to use systems or tools
  - Helping others with high-level design issues
- **Late submission is NOT ALLOWED for all the HW assignments and projects**
  - Be sure the deadline for every assignment and project

Course Policy: Grading

- **Exams (60%): Midterm (30%), Final (30%)**

- **Group projects (30%)**
  - Idea proposal presentation
  - Final presentation

- **Lab assignments (10%)**

Course Policy: Lectures

Flipped learning

- (Online lecture contents) The contents will be uploaded **by the previous Tuesday**
- (Offline lecture) **Every Monday 3:00 PM ~ (Assigned lecture time: from 2PM to 6PM)**
  - Summary (will be given in English)
  - Q&A (Korean questions are also allowed)

Lecture place

- Laboratory: Frontier 507

Course Policy: Attendance

- **Attendance is strongly recommended!**
  - Important points are announced only during the lecture
  - Exam problems are inferred from the lecture
- **But, attendance itself is not included in the score**
- **Attendance confirmation is done by e-class**
  - http://eclass.seoultech.ac.kr

Course Policy: Understanding

- **Encourage questions!**
  - Questions in Korean are allowed
  - After the class, use emails and Q&A in e-class aggressively

- **For the questions that break the flow of class or need to discuss**
  - Please visit me during office hours (or any time if you make reservations)

Course Assumption

- **Required: Basic programming knowledge**
  - Any kind of programming languages – Python, Java, or, C/C++

- **Recommended: Database concepts**

- **This course is designed for ITM students!**

Environments for Practices

Cloudera Big Data Management Environments

- The instance for the virtual machine will be provided

Environments for Developing Web Services

- Apache Web server, PHP, and MySQL Database
- Total package version for Windows can be found here: http://www.wampserver.com/en/

Textbooks

- **The following textbooks are recommended, but are not necessary**
  - [1] is recommended for big data management systems
  - [2] and [3] are recommended for Web programming, but you have sufficient open sourced resources in Web

[1] Tom White, Hadoop: The Definitive Guide, O’Reilly

[2] Programming PHP, Rasmus Lerdorf, Kevin Tatroe, and Peter MacIntyre, O'Reilly Media Inc.

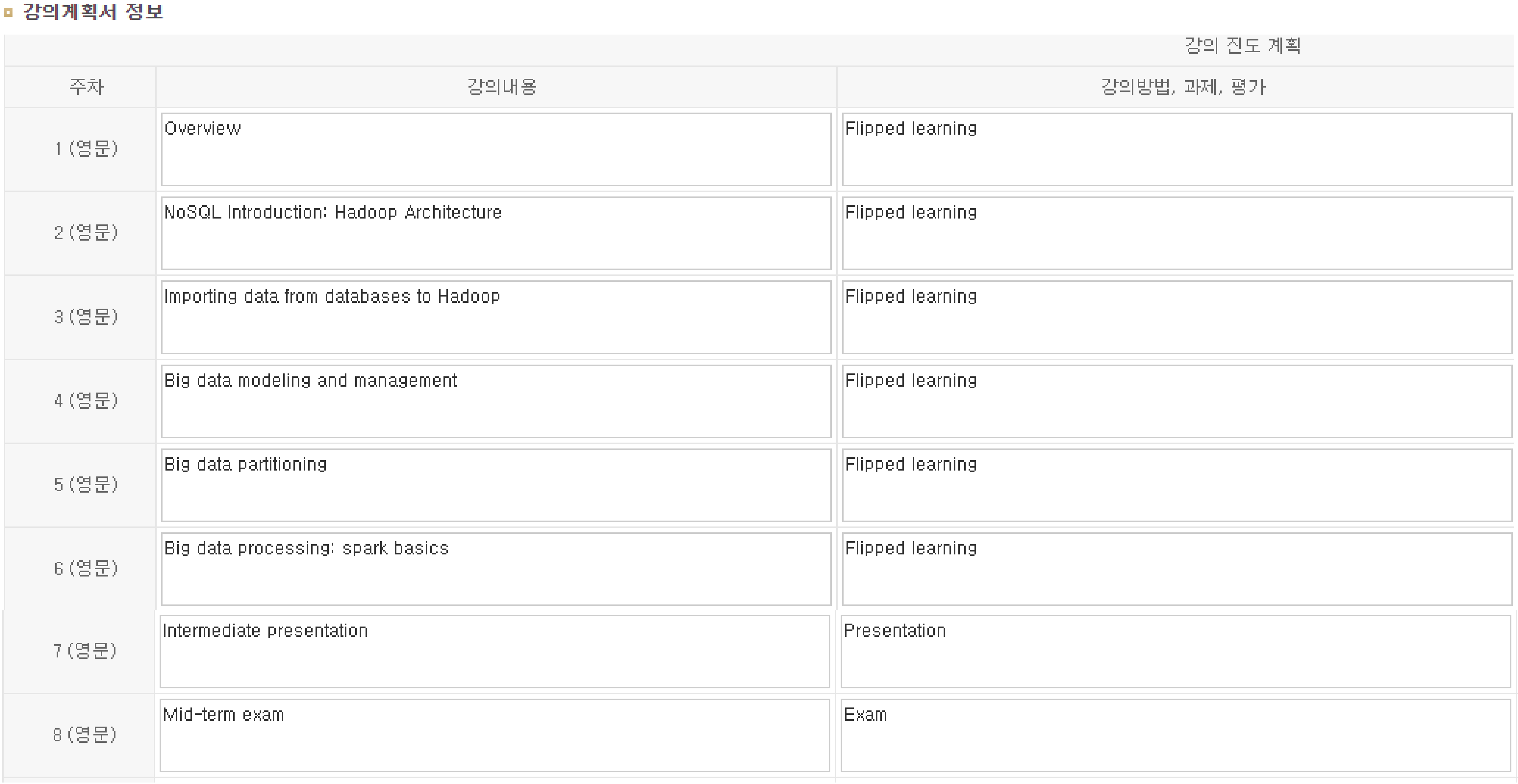
[3] Learning PHP, MySQL & JavaScript, Robin Nixon, O'Reilly Media Inc.

Getting Help

- **E-class: http://eclass.seoultech.ac.kr**
  - Complete schedule of lectures, exams, and assignments
  - Lecture slides, assignments, exams, solutions
  - Clarifications to assignments
  - Q&A

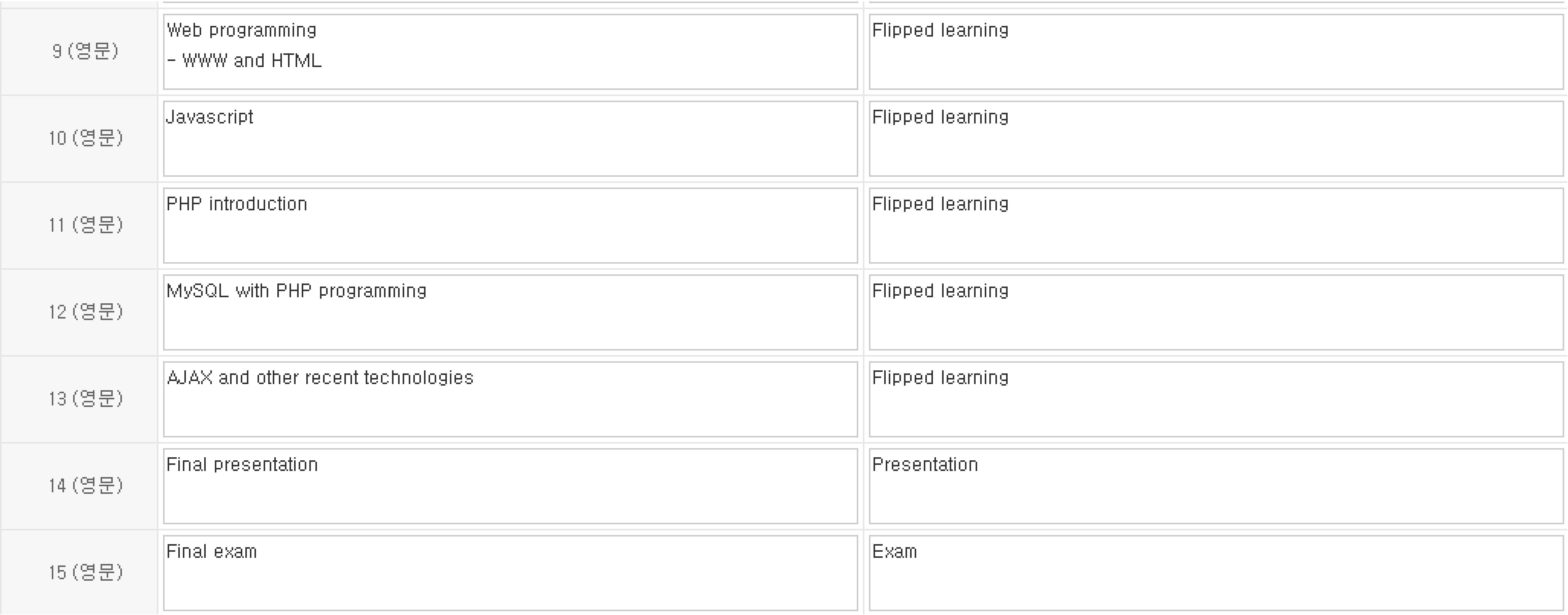
- **Office hours**
  - Wednesday, 2:00-3:00pm, Frontier 614
  - You can schedule 1:1 appointment in advance even not for the office hours

Course Schedule



▫ 강의계획서 정보

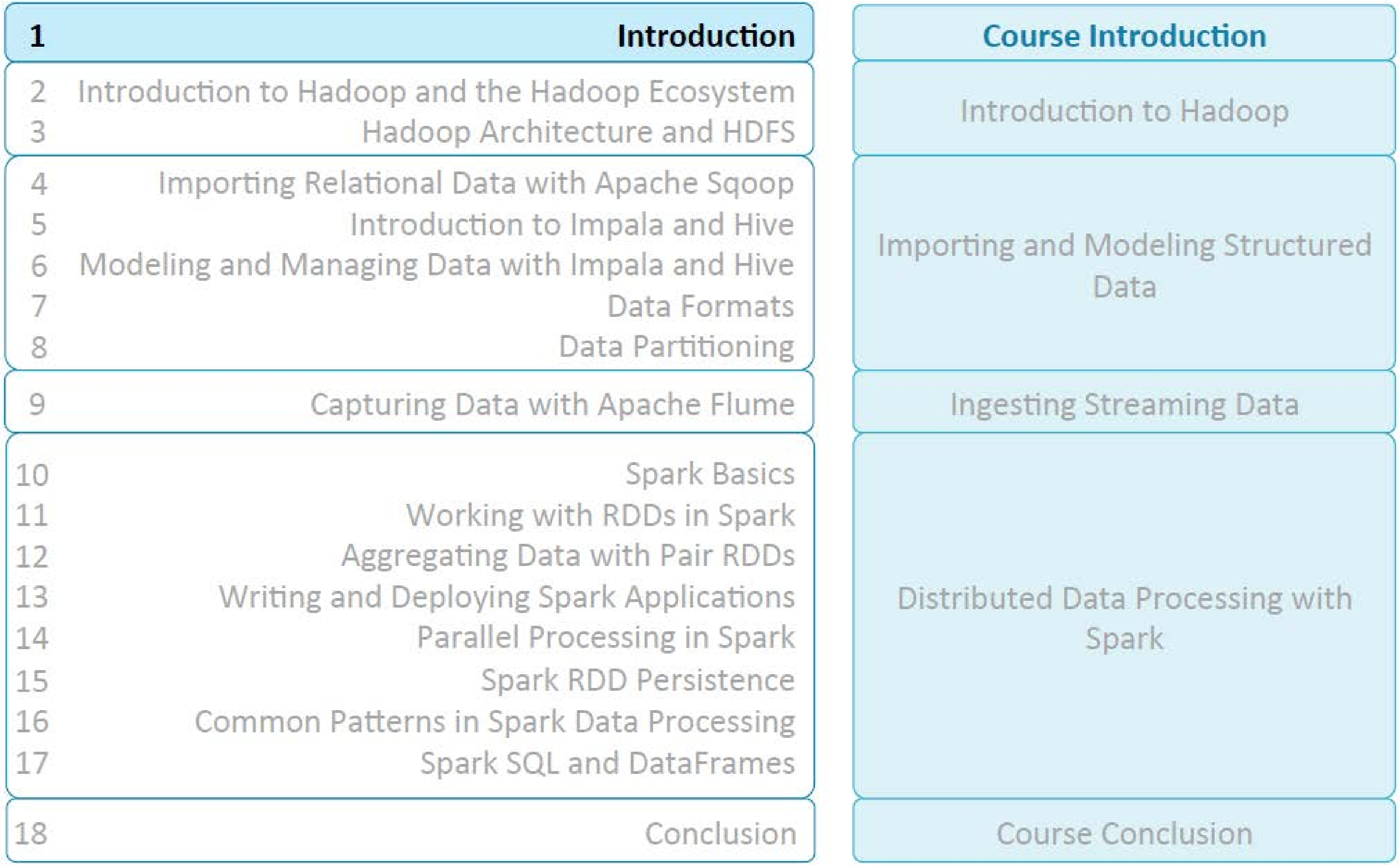
	강의 진도 계획	
주차	강의내용	강의방법, 과제, 평가
1 (영문)	Overview	Flipped learning
2 (영문)	NoSQL Introduction: Hadoop Architecture	Flipped learning
3 (영문)	Importing data from databases to Hadoop	Flipped learning
4 (영문)	Big data modeling and management	Flipped learning
5 (영문)	Big data partitioning	Flipped learning
6 (영문)	Big data processing: spark basics	Flipped learning
7 (영문)	Intermediate presentation	Presentation
8 (영문)	Mid-term exam	Exam



9 (영문)	Web programming - WWW and HTML	Flipped learning
10 (영문)	Javascript	Flipped learning
11 (영문)	PHP introduction	Flipped learning
12 (영문)	MySQL with PHP programming	Flipped learning
13 (영문)	AJAX and other recent technologies	Flipped learning
14 (영문)	Final presentation	Presentation
15 (영문)	Final exam	Exam

Introduction to Big Data Management Systems

Course contents



1	**Introduction**	**Course Introduction**
2	Introduction to Hadoop and the Hadoop Ecosystem	Introduction to Hadoop
3	Hadoop Architecture and HDFS	
4	Importing Relational Data with Apache Sqoop	Importing and Modeling Structured Data
5	Introduction to Impala and Hive	
6	Modeling and Managing Data with Impala and Hive	
7	Data Formats	
8	Data Partitioning	
9	Capturing Data with Apache Flume	Ingesting Streaming Data
10	Spark Basics	Distributed Data Processing with Spark
11	Working with RDDs in Spark	
12	Aggregating Data with Pair RDDs	
13	Writing and Deploying Spark Applications	
14	Parallel Processing in Spark	
15	Spark RDD Persistence	
16	Common Patterns in Spark Data Processing	
17	Spark SQL and DataFrames	
18	Conclusion	Course Conclusion

Course objectives

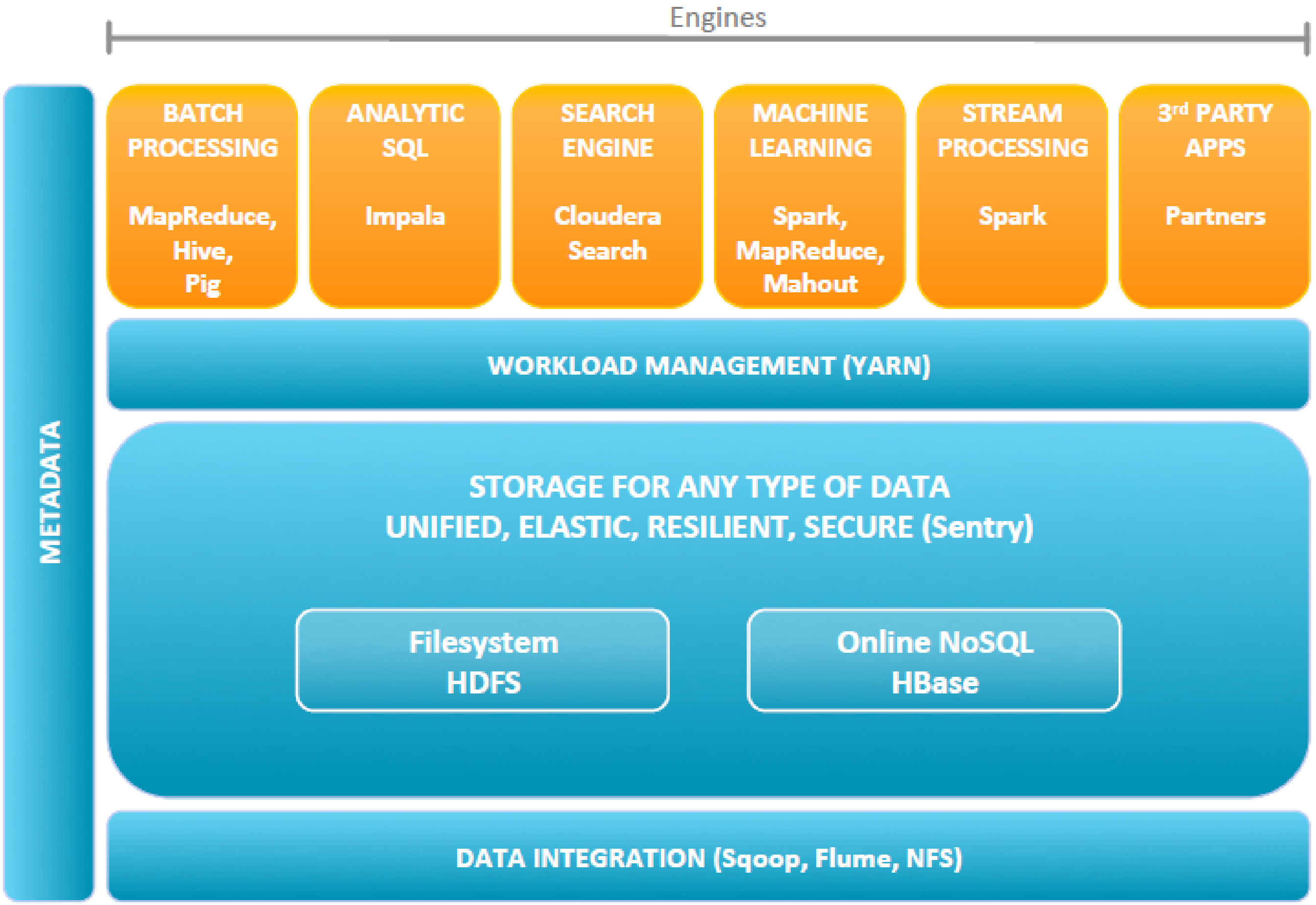
During this course, you will learn

- How the Hadoop Ecosystem fits in with the data processing lifecycle
- How data is distributed, stored and processed in a Hadoop cluster
- How to use Sqoop and Flume to ingest data
- How to process distributed data with Spark
- Best practices for data storage
- How to model structured data as tables in Impala and Hive
- How to choose a data storage format for your data usage patterns

CDH

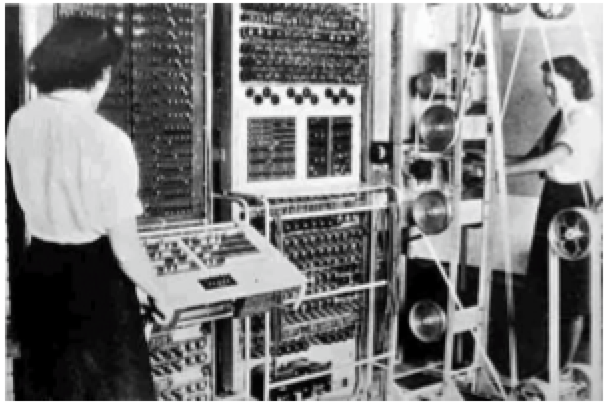
CDH (Cloudera's Distribution including Apache Hadoop)

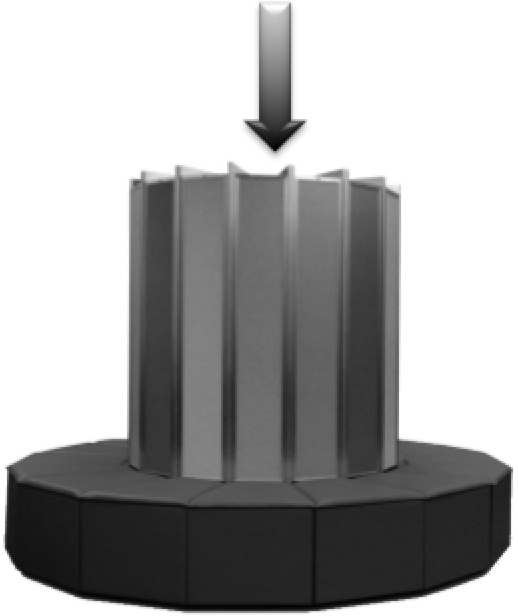
- 100% open source, enterprise-ready distribution of Hadoop and related projects
- The most complete, tested, and widely-deployed distribution of Hadoop
- Integrates all the key Hadoop ecosystem projects
- Available as RPMs and Ubuntu, Debian, or SuSE packages, or as a tarball



Traditional Large-Scale Computation

- **Traditionally, computation has been processor-bound**
  - Relatively small amounts of data
  - Lots of complex processing



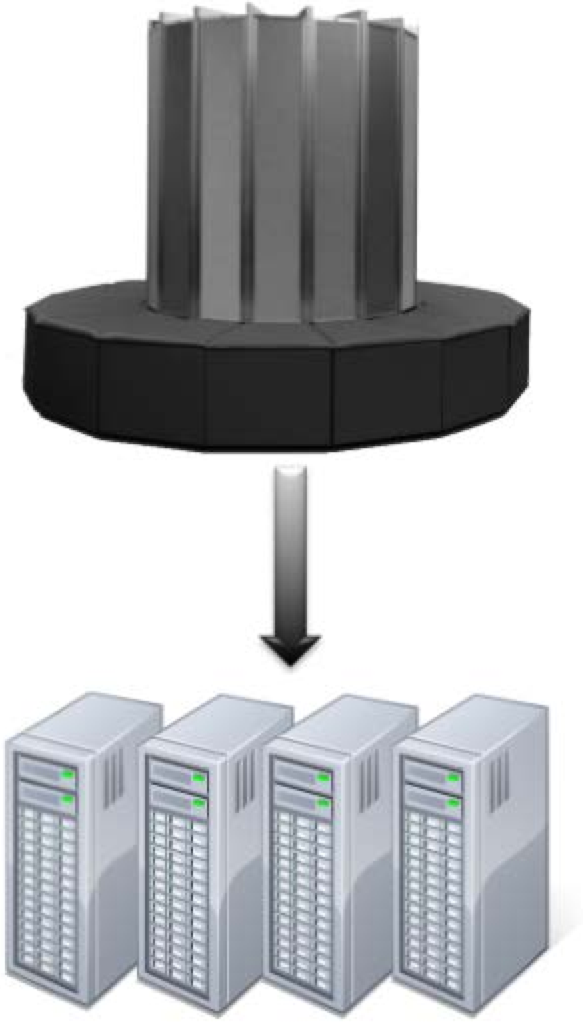


- **The early solution: bigger computers**
  - Faster processor, more memory
  - But even this couldn't keep up

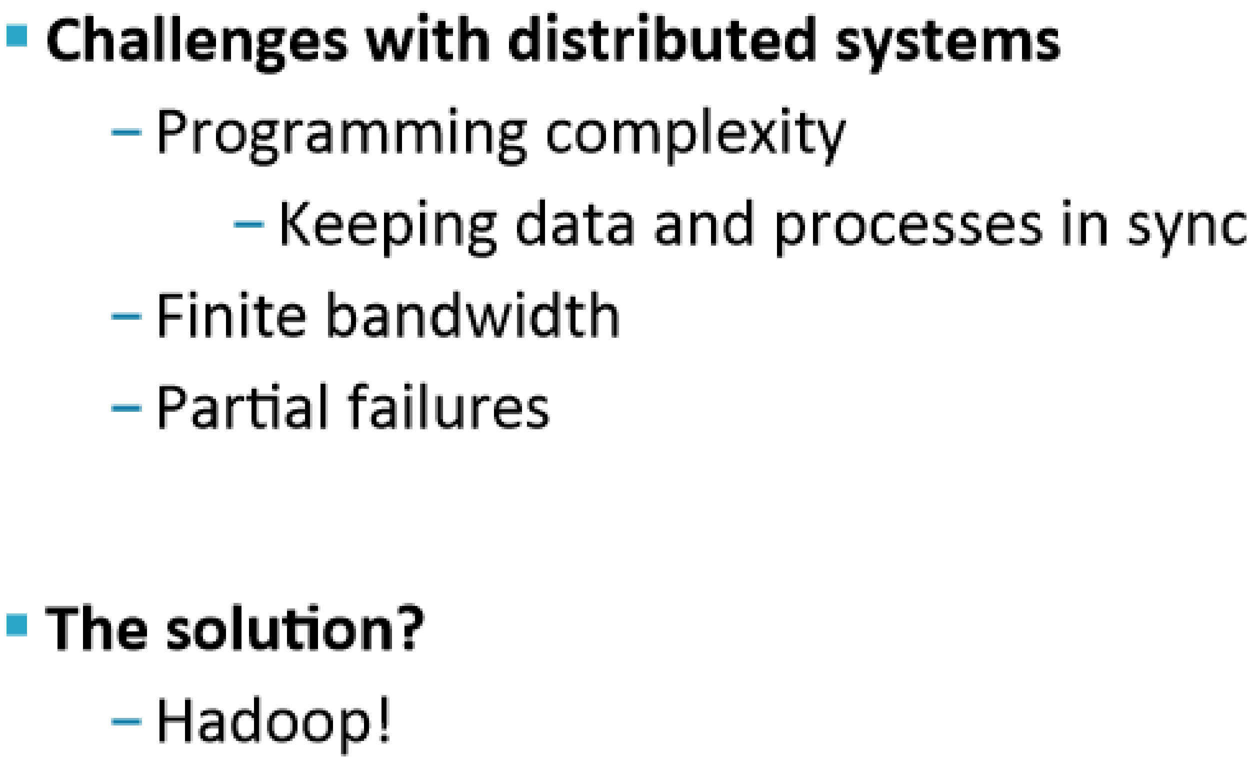
Distributed Systems

- **The better solution: more computers**
  - Distributed systems – use multiple machines for a single job

> “In pioneer days they used oxen for heavy pulling, and when one ox couldn’t budge a log, we didn’t try to grow a larger ox. We shouldn’t be trying for bigger computers, but for *more systems* of computers.”
>
> – Grace Hopper



Challenges with Distributed Systems

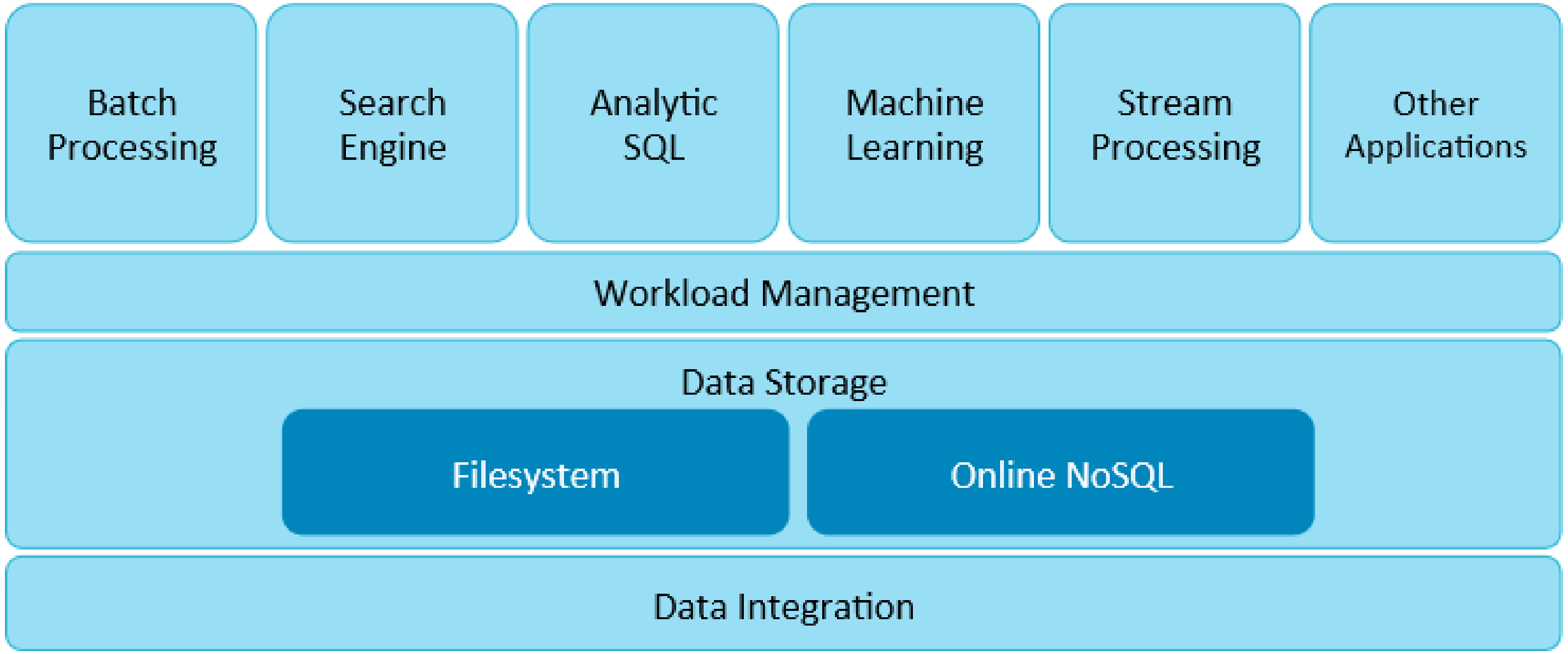


- **Challenges with distributed systems**
  - Programming complexity
    - Keeping data and processes in sync
  - Finite bandwidth
  - Partial failures

- **The solution?**
  - Hadoop!

What is Apache Hadoop?

- **Scalable and economical data storage, processing and analysis**
  - Distributed and fault-tolerant
  - Harnesses the power of industry standard hardware
- **Heavily inspired by technical documents published by Google**



Batch Processing

Search Engine

Analytic SQL

Machine Learning

Stream Processing

Other Applications

Workload Management

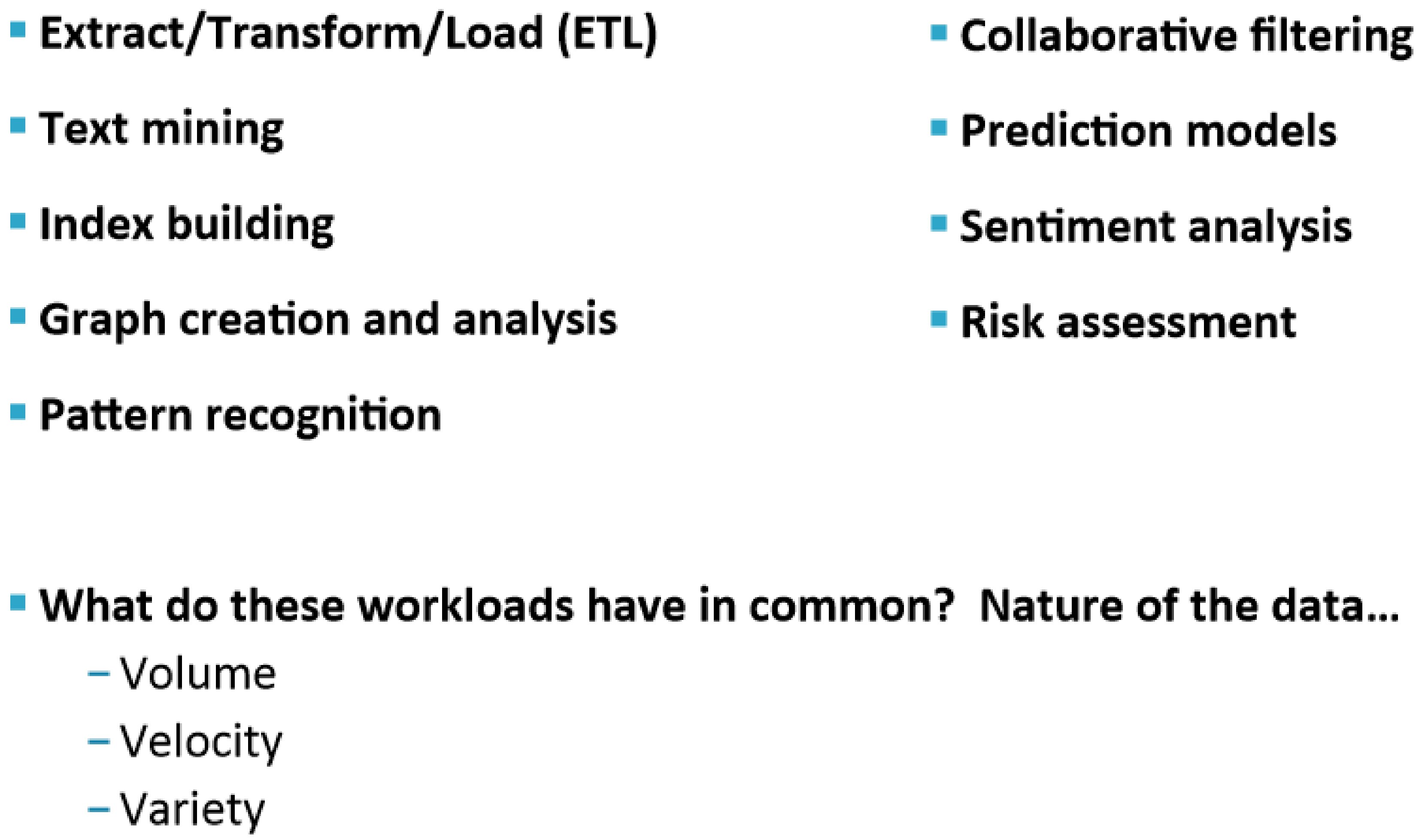
Data Storage

Filesystem

Online NoSQL

Data Integration

Common Hadoop Use Cases

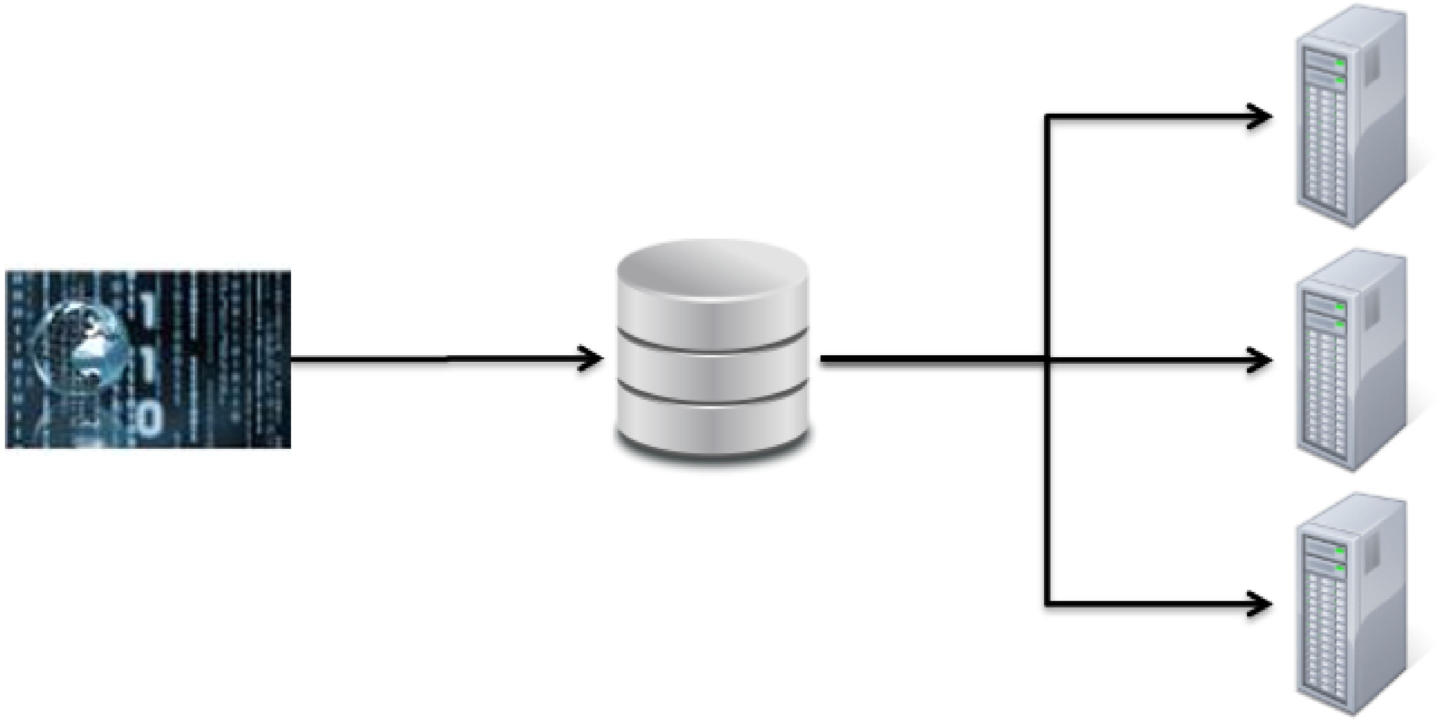


- Extract/Transform/Load (ETL)
- Text mining
- Index building
- Graph creation and analysis
- Pattern recognition
- Collaborative filtering
- Prediction models
- Sentiment analysis
- Risk assessment

- What do these workloads have in common? Nature of the data…
  - Volume
  - Velocity
  - Variety

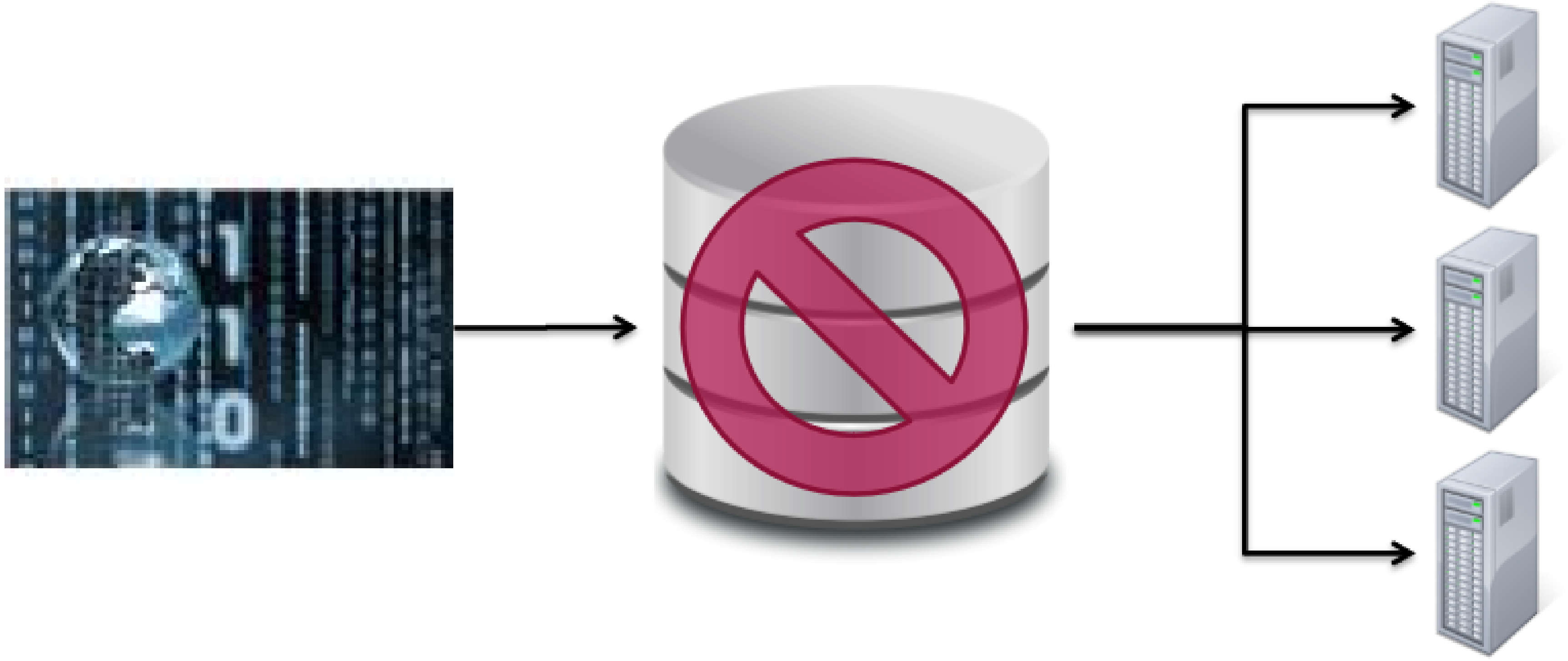
Distributed Systems: The Data Bottleneck (1)

- Traditionally, data is stored in a central location
- Data is copied to processors at runtime
- Fine for limited amounts of data



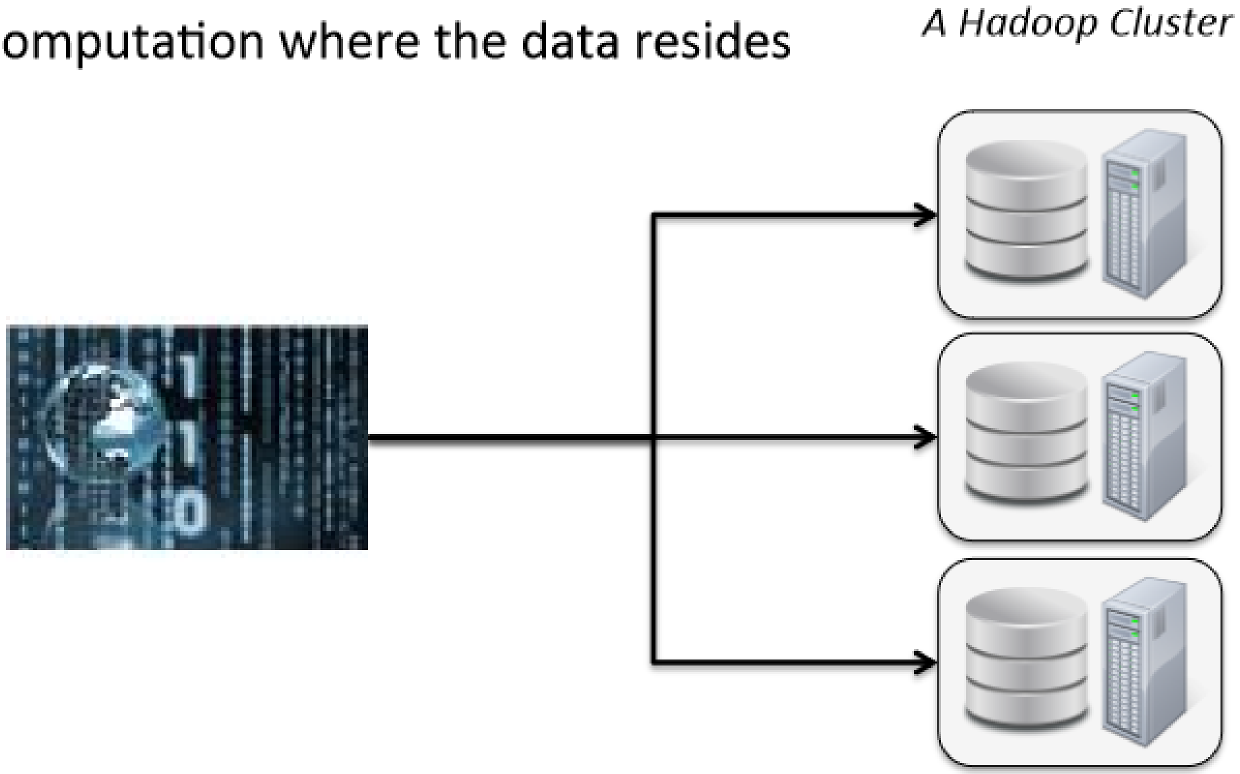
Distributed Systems: The Data Bottleneck (2)

- **Modern systems have much more data**
  - terabytes+ a day
  - petabytes+ total
- **We need a new approach…**

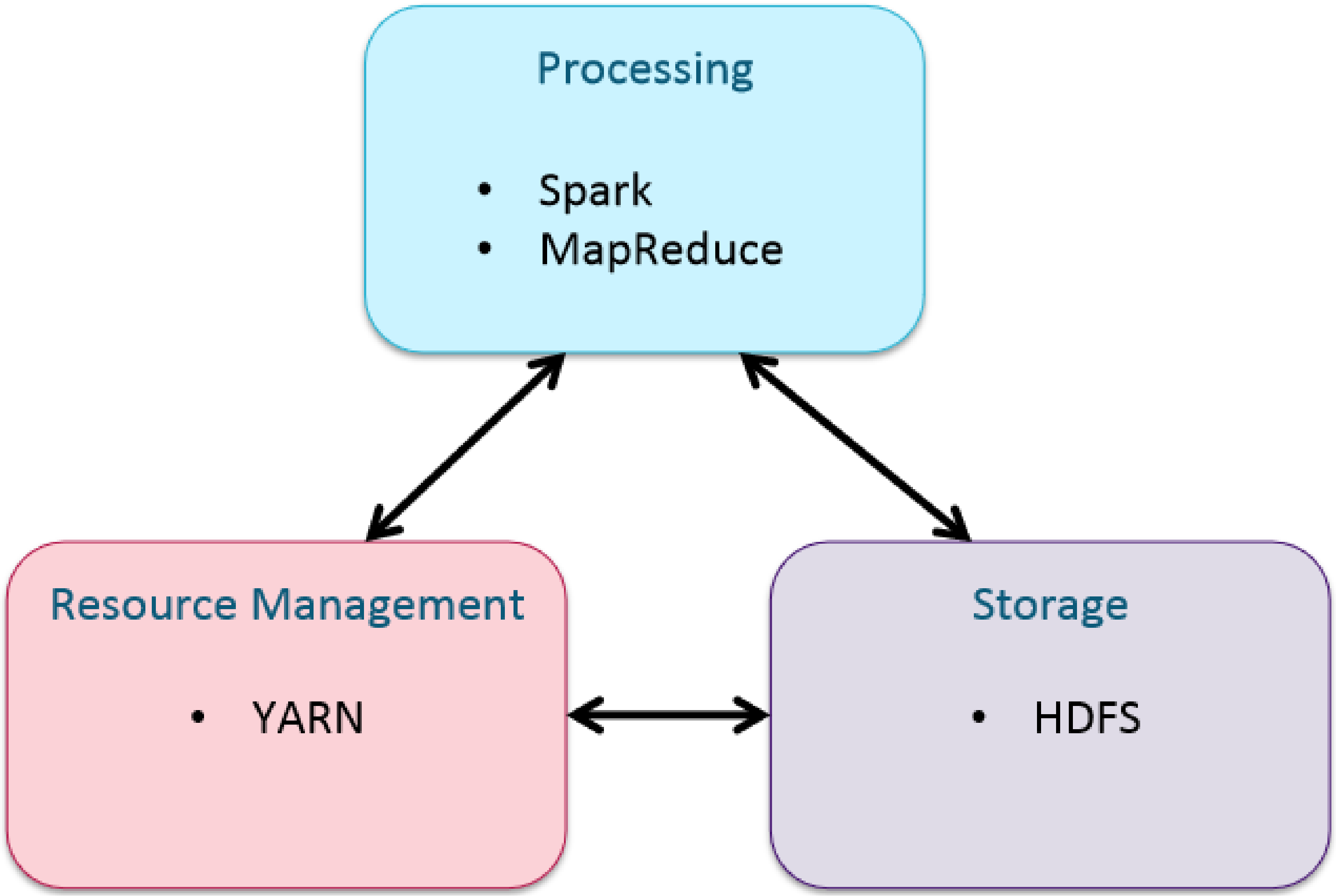


Big Data Processing with Hadoop

- **Hadoop introduced a radical new approach:**
  - Bring the program to the data rather than the data to the program
- **Based on two key concepts**
  - Distribute data when the data is stored
  - Run computation where the data resides



Core Hadoop



Processing

- Spark
- MapReduce

Resource Management

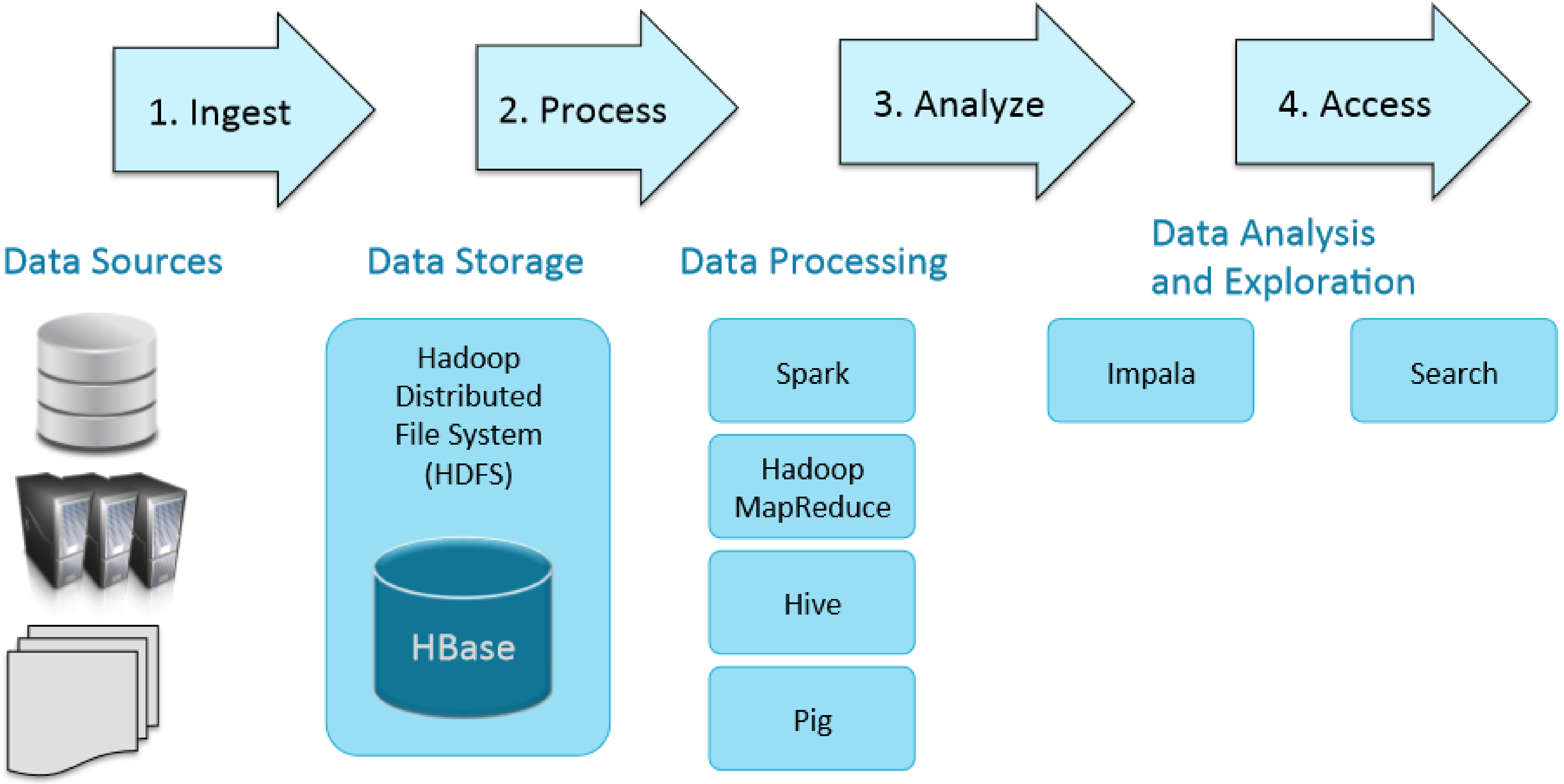
- YARN

Storage

- HDFS

A Hadoop Cluster

Big Data Processing



1. Ingest

2. Process

3. Analyze

4. Access

Data Sources

Data Storage

Data Processing

Data Analysis and Exploration

Hadoop Distributed File System (HDFS)

HBase

Spark

Hadoop MapReduce

Hive

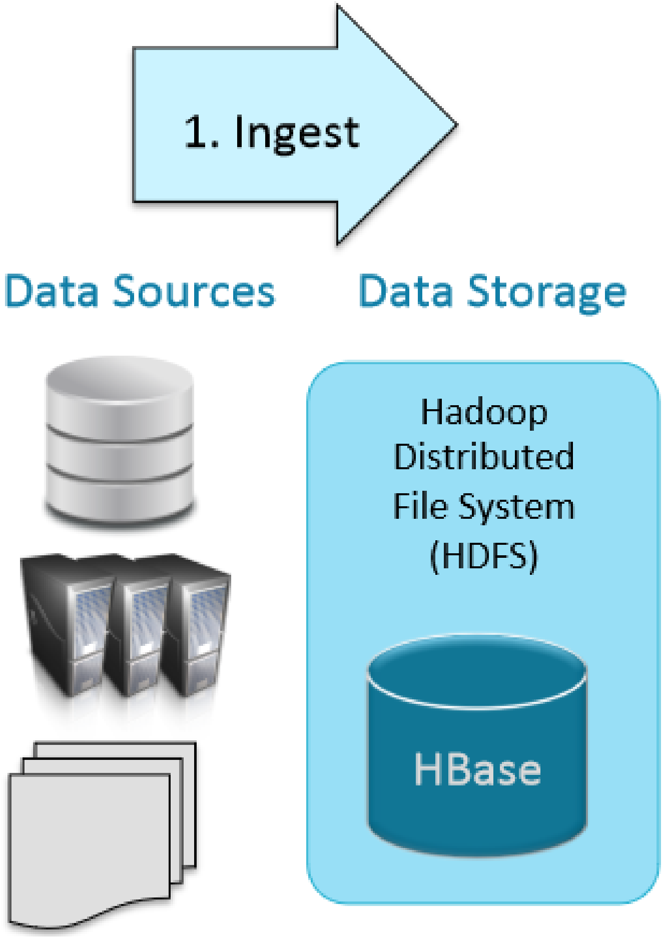
Pig

Impala

Search

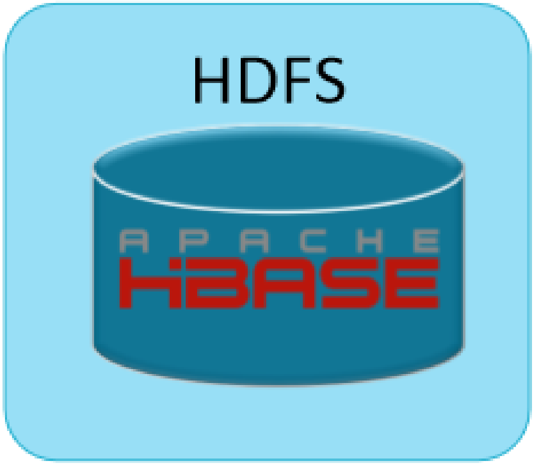
Data Ingest and Storage

- **Hadoop typically ingests data from many sources and in many formats**
  - Traditional data management systems, e.g. databases
  - Logs and other machine generated data (event data)
  - Imported files



Data Storage

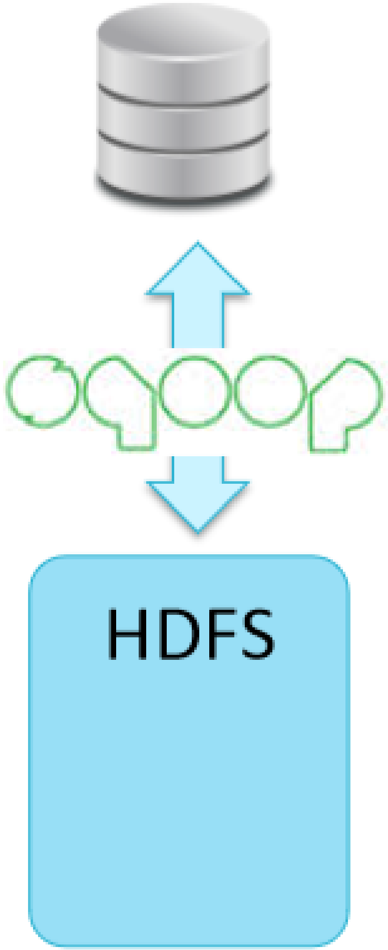
- **Hadoop Distributed File System (HDFS)**
  - HDFS is the storage layer for Hadoop
  - Provides inexpensive reliable storage for massive amounts of data on industry-standard hardware
  - Data is distributed when stored
  - Covered later in this course



- **Apache HBase: The Hadoop Database**
  - A NoSQL distributed database built on HDFS
  - Scales to support very large amounts of data and high throughput
  - A table can have thousands of columns
  - Covered in depth in *Cloudera Training for Apache HBase*

Data Ingest Tools (1)

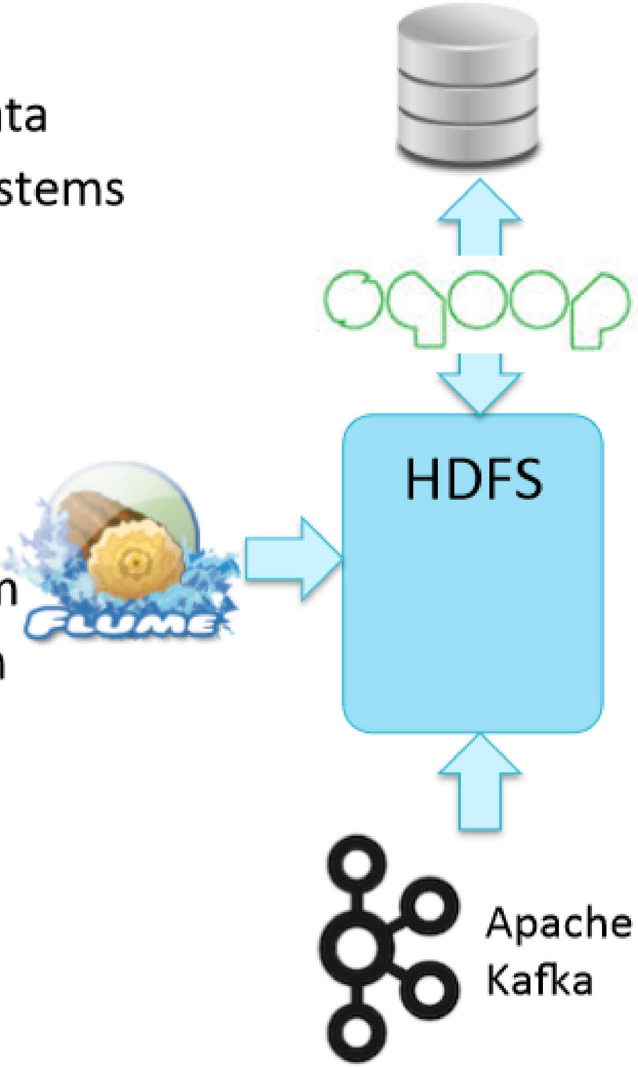
- **HDFS**
  - Direct file transfer
- **Apache Sqoop**
  - High speed import to HDFS from Relationship Database (and vice versa)
  - Supports many data storage systems
    - e.g. Netezza, Mongo, MySQL, Teradata, Oracle
  - Covered later in this course



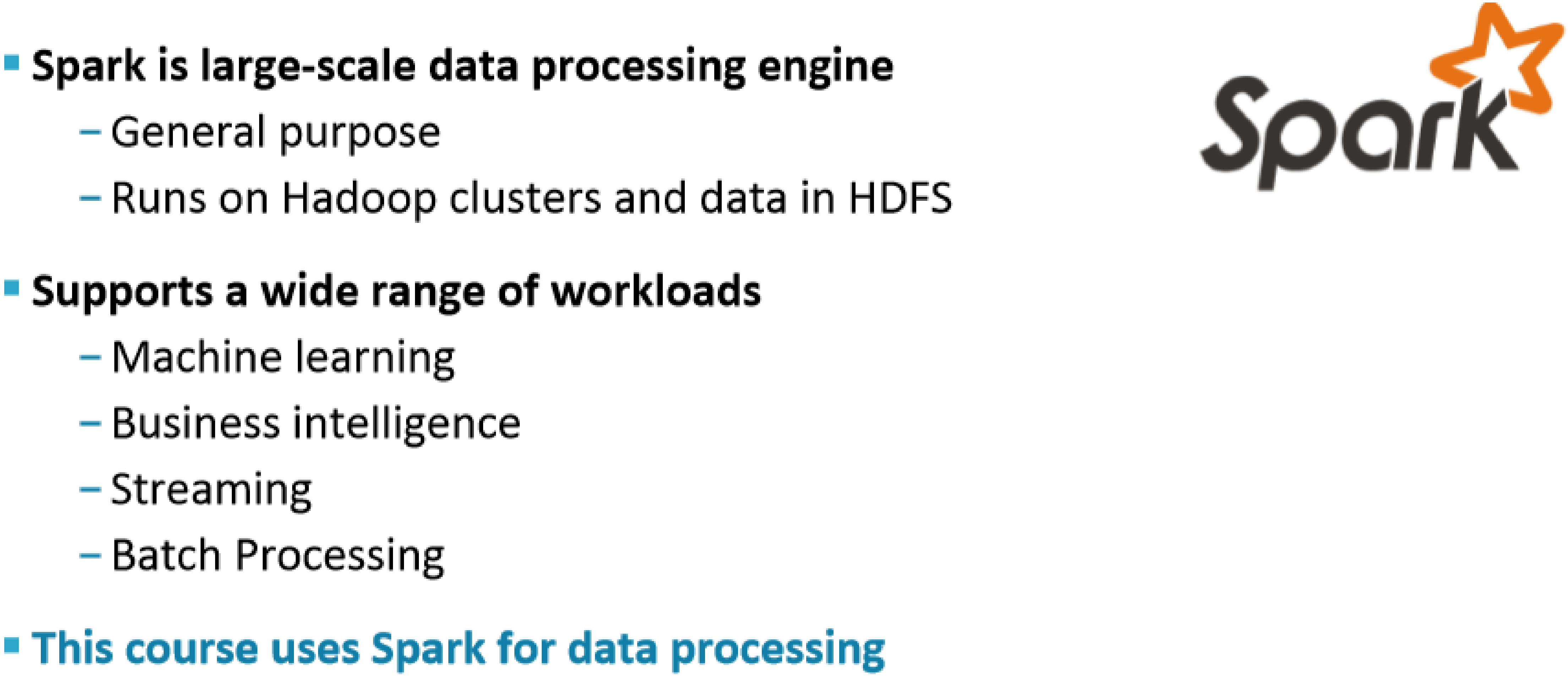
Data Ingest Tools (2)

- **Apache Flume**
  - Distributed service for ingesting streaming data
  - Ideally suited for event data from multiple systems
    - For example, log files

- **Kafka**
  - A high throughput, scalable messaging system
  - Distributed, reliable publish-subscribe system
  - Integrates with Flume and Spark Streaming



Apache Spark: An Engine For Large-scale Data Processing



- **Spark is large-scale data processing engine**
  - General purpose
  - Runs on Hadoop clusters and data in HDFS
- **Supports a wide range of workloads**
  - Machine learning
  - Business intelligence
  - Streaming
  - Batch Processing
- **This course uses Spark for data processing**

Hadoop MapReduce: The Original Hadoop Processing Engine

- Hadoop MapReduce is the original Hadoop framework
  - Primarily Java based
- Based on the MapReduce programming model
- The core Hadoop processing engine before Spark was introduced
- Still the dominant technology
  - But losing ground to Spark fast
- Many existing tools are still built using MapReduce code
- Has extensive and mature fault tolerance built into the framework

Apache Pig: Scripting for MapReduce

- **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

- **The Pig interpreter runs on the client machine**
  - Turns Pig Latin scripts into MapReduce or Spark jobs
  - Submits those jobs to a Hadoop cluster
  - Covered in Cloudera *Data Analyst Training*

```
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;
```

Impala: High Performance SQL

- **Impala is a high-performance SQL engine**
  - Runs on Hadoop clusters
  - Data stored in HDFS files
  - Inspired by Google's Dremel project
  - Very low latency – measured in milliseconds
  - Ideal for interactive analysis
- **Impala supports a dialect of SQL (Impala SQL)**
  - Data in HDFS modeled as database tables
- **Impala was developed by Cloudera**
  - 100% open source, released under the Apache software license
- **Impala is used for data analysis in this course**

Apache Hive: SQL on MapReduce

- **Hive is an abstraction layer on top of Hadoop**
  - Hive uses a SQL-like language called HiveQL
  - Similar to Impala SQL
  - Useful for data processing and ETL
    - Impala is preferred for ad hoc analytics

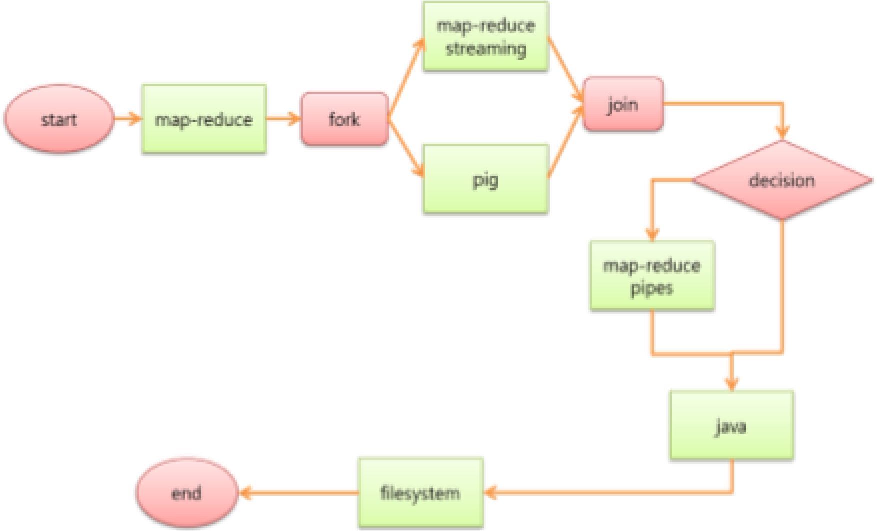
- **Hive executes queries using MapReduce**
  - Hive on Spark is available for early adopters; not yet recommended for production
- **Hive can optionally be used for data analysis in this course**

Hue: The UI for Hadoop

- **Hue = Hadoop User Experience**
- **Hue provides a Web front-end to a Hadoop**
  - Upload and browse data
  - Query tables in Impala and Hive
  - Run Spark and Pig jobs and workflows
  - Search
  - And much more
- **Makes Hadoop easier to use**
- **Hue is 100% open-source**
- **Created by Cloudera**
  - Open source, released under Apache license
- **Hue is used throughout this course**

Apache Oozie: Workflow Management

- **Oozie**
  - Workflow engine for Hadoop jobs
  - Defines dependencies between jobs
- **The Oozie server submits the jobs to the server in the correct sequence**



Apache Sentry: Hadoop Security

- **Sentry provides fine-grained access control (authorization) to various Hadoop ecosystem components**
  - Impala
  - Hive
  - Cloudera Search
  - HDFS
- **In conjunction with Kerberos authentication, Sentry authorization provides a complete cluster security solution**
- **Created by Cloudera**
  - Now an open-source Apache project

Summary

- **Hadoop is a framework for distributed storage and processing**
- **Core Hadoop includes HDFS for storage and YARN for cluster resource management**
- **The Hadoop ecosystem includes many components for**
  - Ingesting data (Flume, Sqoop, Kafka)
  - Storing data (HDFS, HBase)
  - Processing data (Spark, Hadoop MapReduce, Pig)
  - Modeling data as tables for SQL access (Impala, Hive)
  - Exploring data (Hue, Search)
  - Protecting Data (Sentry)
- **This course introduces most of the key Hadoop infrastructure**

Welcome and Enjoy!