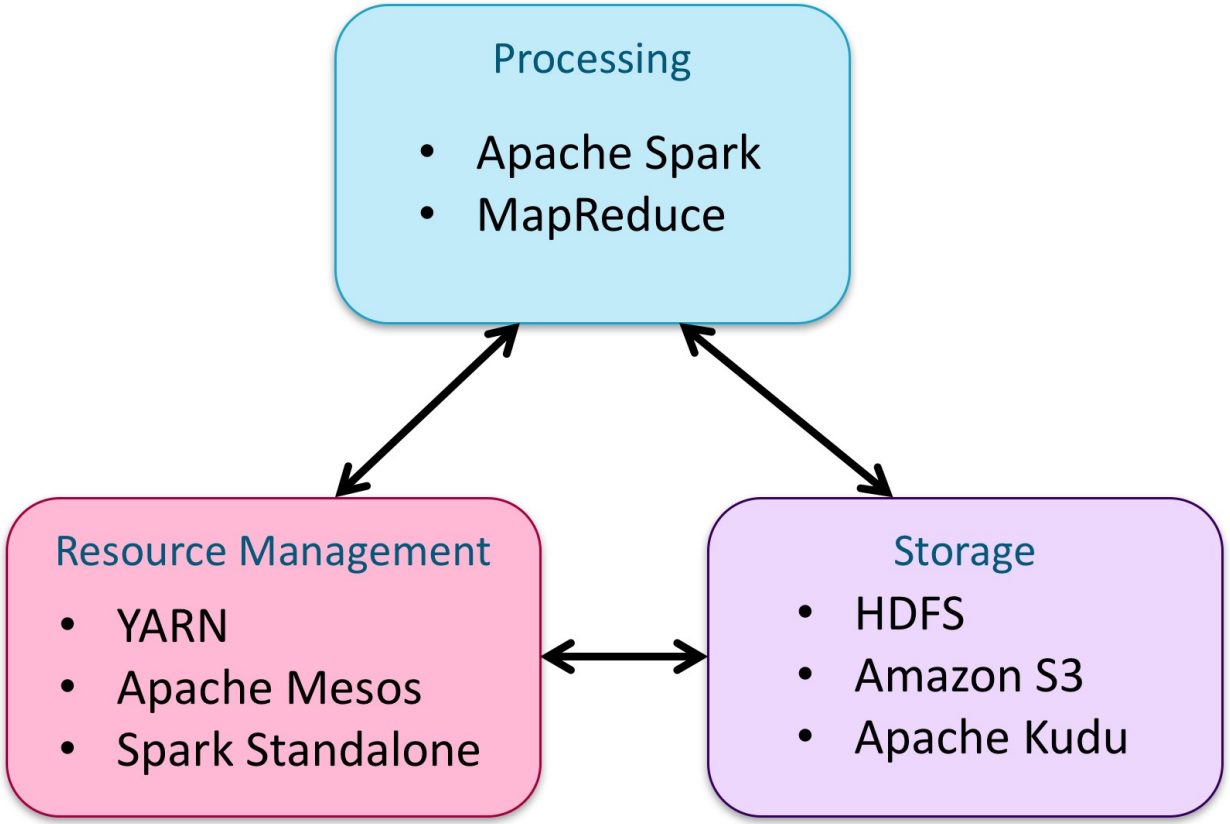
Introduction

Distributed Processing with Hadoop



A Hadoop Cluster

Apache Spark: An Engine for Large-Scale Data Processing

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

Hadoop MapReduce: The Original Hadoop Processing Engine

- **Hadoop MapReduce is the original Hadoop framework for processing big data**
  - Primarily Java-based
- **Based on the map-reduce programming model**
- **The core Hadoop processing engine before Spark was introduced**
- **Still in use in many production systems**
  - 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**

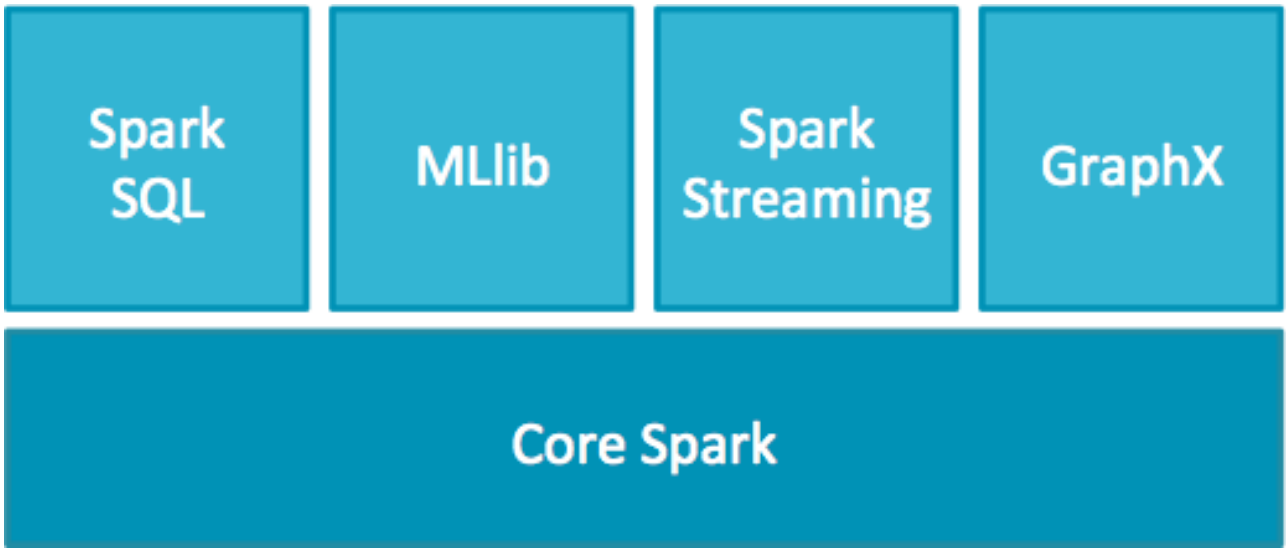
Apache Spark Basics

What Is Apache Spark?

- **Apache Spark is a fast and general engine for large-scale data processing**
- **Written in Scala**
  - Functional programming language that runs in a JVM
- **Spark shell**
  - Interactive—for learning, data exploration, or ad hoc analytics
  - Python or Scala
- **Spark applications**
  - For large scale data processing
  - Python, Scala, or Java

The Spark Stack

- **Spark provides a stack of libraries built on core Spark**
  - Core Spark provides the fundamental Spark abstraction: Resilient Distributed Datasets (RDDs)
  - Spark SQL works with structured data
  - MLlib supports scalable machine learning
  - Spark Streaming applications process data in real time
  - GraphX works with graphs and graph-parallel computation



Spark SQL

- **Spark SQL is a Spark library for working with structured data**
- **What does Spark SQL provide?**
  - The DataFrame and Dataset API
    - The primary entry point for developing Spark applications
    - DataFrames and Datasets are abstractions for representing structured data
  - Catalyst Optimizer—an extensible optimization framework
  - A SQL engine and command line interface

The Spark Shell

- **The Spark shell provides an interactive Spark environment**
  - Often called a *REPL*, or Read/Evaluate/Print Loop
  - For learning, testing, data exploration, or ad hoc analytics
  - You can run the Spark shell using either Python or Scala
- **You typically run the Spark shell on a gateway node**

Starting the Spark Shell

- **On a Cloudera cluster, the command to start the Spark 2 shell is**
  - `pyspark` for Python
  - `spark-shell` for Scala
- **The Spark shell has a number of different start-up options, including**
  - `master`: specify the cluster to connect to
  - `jars`: Additional JAR files (Scala only)
  - `py-files`: Additional Python files (Python only)
  - `name`: the name the Spark Application UI uses for this application
    - Defaults to `PySparkShell` (Python) or `Spark shell` (Scala)
  - `help`: Show all the available shell options

```
$ pyspark --name "My Application"
```

Lab : Install Spark in centos Linux. (45 Minutes)