Cloud Computing 3

CSCI 4850/5850 High-Performance Computing

Spring 2018

Tae-Hyuk (Ted) Ahn

Department of Computer Science
Program of Bioinformatics and Computational Biology
Saint Louis University
Learning Objectives

- You will learn about how to use Apache Spark
Apache Spark

In-Memory Cluster Computing for Iterative and Interactive Applications

• Apache Spark is an open-source cluster computing framework for real-time processing.
• It is one of the most successful projects in the Apache Software Foundation.
• Spark has clearly evolved as the market leader for Big Data processing.
Motivation

- Current popular programming models for clusters transform data flowing from stable storage to stable storage
- E.g., MapReduce:
Motivation

- Acyclic data flow is a powerful abstraction, but is not efficient for applications that repeatedly reuse a *working set* of data:
  - **Iterative** algorithms (many in machine learning)
  - **Interactive** data mining tools (R, Excel, Python)
- Spark makes working sets a first-class concept to efficiently support these apps
Why Spark when Hadoop is already there?

- Move data through disk and network (HDFS)
- User can cache data in memory

[Source](https://acadgild.com/blog/hadoop-vs-spark-best-big-data-frameworks/)
Spark Goal

- Provide distributed memory abstractions for clusters to support apps with working sets
- Retain the attractive properties of MapReduce:
  - Fault tolerance (for crashes & stragglers)
  - Data locality
  - Scalability

**Solution:** augment data flow model with “resilient distributed datasets” (RDDs)
Spark

**Speed**
Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

![Running time comparison chart](chart.png)

**Ease of Use**
Write applications quickly in Java, Scala, Python, R.

- Spark SQL
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)

Apache Spark
Spark Features: Polyglot

- Spark provides high-level APIs in Java, Scala, Python and R.
- Spark code can be written in any of these four languages.
- It provides a shell in Scala and Python.
- The Scala shell can be accessed through ./bin/spark-shell and Python shell through ./bin/pyspark from the installed directory.
Spark Features: Multiple Formats

- Spark supports multiple data sources such as Parquet, JSON, Hive and Cassandra apart from the usual formats such as text files, CSV and RDBMS tables.
- The Data Source API provides a pluggable mechanism for accessing structured data through Spark SQL.
- Data sources can be more than just simple pipes that convert data and pull it into Spark.
Spark Features: Real Time Computation

- Spark’s computation is real-time and has low latency because of its in-memory computation.
- Spark is designed for massive scalability and the Spark team has documented users of the system running production clusters with thousands of nodes and supports several computational models.
Spark Features: Hadoop Integration

- Apache Spark provides smooth compatibility with Hadoop.
- This is a boon for all the Big Data engineers who started their careers with Hadoop.
- Spark is a potential replacement for the MapReduce functions of Hadoop, while Spark has the ability to run on top of an existing Hadoop cluster using YARN for resource scheduling.
Spark Features: Machine Learning

- Spark’s MLlib is the machine learning component which is handy when it comes to big data processing.
- It eradicates the need to use multiple tools, one for processing and one for machine learning.
- Spark provides data engineers and data scientists with a powerful, unified engine that is both fast and easy to use.
Apache Spark on Amazon

- Install Apache Spark in your Desktop or Cluster
  - https://spark.apache.org/docs/latest/
- Spark Amazon EC2: scripts that let you launch a cluster on EC2
  - https://github.com/amplab/spark-ec2
- Amazon EMR
Lab

- https://spark.apache.org/docs/latest/quick-start.html