Efficient and developer-friendly machine learning development on Apache Spark

Presented by Kenneth Lui (wckl2)
1st December, 2015

Agenda

- Why Spark?
- Motivation
- Project Plan
- Deliverable

Why Spark?

Active Developer Community

- GitHub
  - 750+ Contributors
  - 6,400+ Stars
- Online courses
  - edX - Introduction to Big Data with Apache Spark
  - DataStax - DataStax Enterprise Analytics with Apache Spark
- Spin-off company
  - Databricks

Availability

- AWS Elastic MapReduce 4.x supports Spark
- Databricks Cloud

Built on top of Spark

- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)
- 150+ Spark packages
    - E.g. **adam**: A genomics processing engine and specialized file format built using Apache Avro, Apache Spark and Parquet
    - or **Thunder**: Large-scale image and time series analysis with Spark

Motivation

Pain Point

- Learning a distributed system takes longer than it should be
  - Resolving dependency
  - Configuration
- People get frustrated by the irrelavant troubleshooting before they can start any coding

Solution

- Address the deployment part of learning Spark
  - Major obstacle for beginners
- Transparent and incremental approach for developers to learn
- A demo project showcasing different options in using Spark and how they affect the performance

Project Plan

Project Plan

1. Installation
    a. Binary v.s. Build from source
    b. Understand the dependency
        i. E.g. how it works with different Hadoop version, HDFS etc.

Project Plan

2. Deployment
    a. Spark’s own standalone cluster manager v.s. Mesos v.s YARN
    b. Qualitative comparison between these three choices
    c. Cloud-based deployment on Amazon Web Services

Project Plan

3. Applying basic machine learning techniques
    a. Native Spark
    b. MLLib
        i. Classification: SVMWithSGD, LogisticRegressionWithSGD
        ii. Regression: LinearRegressionWithSGD, RidgeRegressionWithSGD and LassoWithSGD

Project Plan

4. Performance evaluation
    a. Different partition strategy
    b. Number of Executors per Worker Node
    c. Shuffle Behaviour configuration (Buffer size, number of connections etc.)
    d. Serializer (JavaSerializer v.s. KryoSerializer etc.)

Deliverable

- A detailed tutorial for setting up Spark in various environments
- Multiple examples for applying basic machine learning techniques
- Implement some custom machine learning algorithms
- Performance evaluation demonstrating different partition strategies
- Everything available on GitHub as an open source project

Questions?