INTRODUCTION TO APACHE SPARK

Jose Nandez, PhD
jnandez@sharcnet.ca
Big Data Specialist
SHARCNET/Western University
June 2016
WHAT IS APACHE SPARK?

- Apache Spark or (just Spark) is a fast and general engine for processing large-scale datasets.
- Spark extends the MapReduce model, supporting interactive queries and stream processing.
- Spark has the ability to run computations in memory or disk (MapReduce) depending on the complexity of the problem.
- Spark is designed to work on batch applications, iterative algorithms, interactive queries, and streaming.
LITTLE HISTORY OF SPARK

- Spark is open source
- Spark started in 2009 as a research project in UC Berkeley RAD Lab.
- Researchers there realised that Hadoop MapReduce was inefficient for interactive and iterative computing jobs
- Papers show that Spark is 10-20x faster than MapReduce in 2009
- In March 2010, Spark became open source
- In June 2013, Spark was accepted in the Apache Software Foundation
- Now, there are some paper claiming up to 100x faster than MapReduce
ACCESSING SPARK

- Spark is highly accessible, offering few API for
  - Python
  - Scala
  - Java
  - SQL
  - R
- Spark is written in Scala and
- Scala is written in Java, therefore Spark uses JVM
- Current Stable Version 1.6.1, coming soon 2.0.0!
SPARK LIBRARIES

- Spark SQL lets you query structured data
- Spark Streaming lets you ingest live data streams (such as Twitter data)
- MLlib is a scalable machine learning library
- GraphX is for graphs and graph-parallel computation for graph analysis (such as Facebook)
WHERE DOES IT RUN?

- Spark runs on
  - Hadoop (MapReduce Model)
  - Mesos (distributed system kernel)
  - Amazon EC2
  - Standalone (the version that we have in SHARCNET)
  - In a Cloud
- It can access diverse data sources
  - Hadoop Distributed File System (HDFS)
  - Cassandra (database)
  - HBase (Big data store and Hadoop database, also Big Table)
  - Amazon Simple Storage Service (S3)
  - MongoDB
Resilient Distributed Dataset (RDD) is the basic Spark data structure
All work in Spark is expressed in RDDs
RDDs are the core of Spark
RDD is immutable distributed collection of objects
RDDs are distributed by Spark across multiple partitions
RDDs can contain any type of Python, Scala, Java or R objects
SPARK OPERATIONS

- **Transformations:** operations on RDD that return a new RDD (check the demo) such as filtering. Examples
  - Map
  - Filter

- **Actions:** operations that return the final value to the driver program or to the disk
  - Take
  - Collect
Spark uses lazy evaluation on RDD.

Lazy evaluation means that Spark will not execute until an action.

Convenient for reading portion of data.

Loading data is also lazily evaluated! Data not loaded until it is need (an action call).
Spark uses SparkContext to connect to a Spark Cluster

- SparkContext (sc) is always initiated in the interactive mode, but not in a script
- SparkContext can be used to create RDDs on the Spark Cluster
- Only one SparkContext may be active per JVM
- SparkContext is necessary in all Spark applications.
APPLICATIONS OF SPARK

- Data Science
- Recommending Music, Movies or any product (like in Amazon or NetFlix)
- For fraud, detect network attacks using all history and machine learning
- Financial risk with Monte Carlo Simulations
- Analysing friendship (like Facebook) with GraphX
- Finding planets by means of all Kepler data
- Finding patterns in traffic from GPS data and recommend new trips
HOW TO SUBMIT A SPARK JOB

- `ssh username@mosaic.sharcnet.ca`
- `module load python/intel/2.7.8`
- `module load spark`
- `sqsub -r time -o log_file spark-submit script.py, for a serial Python job`
- `sqsub -q threaded -n #CPU -r time -o log_file spark-submit script.py, for a multithreaded Python job`
- `spark-submit is the command used for submitting any Spark script (from Python, Scala, R, Java, SQL)`
DEVELOP SPARK SCRIPT ON SHARCNET

- ssh username@mosaic.sharcnet.ca, or redfin
- ssh mos-dev1, log into the development node
- module load python/intel/2.7.8 #add this to your bashrc
- module load spark #add this to your bashrc
- pyspark  #this will start the Python interactive session
- If you prefer IPython (recommended)
- If you always want IPython, add export IPYTHON=1 to your bashrc
  - IPYTHON=1 pyspark
WHERE TO FIND HELP IN SHARCNET?

- https://www.sharcnet.ca/help/index.php/Apache_Spark
- help@sharcnet.ca
- Or email me (jnandez@sharcnet.ca) if you want to know more about Spark
REFERENCES

- Learning Spark: Lightning-Fast Big Data Analysis By Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia
- Advanced Analytics with Spark Patterns for Learning from Data at Scala By Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills
- http://spark.apache.org/
FUNCTIONS IN SPARK

- **map(function):**
  - Applies a function to each element of the list

- **Filter(function):**
  - Applies a function to each element of the list and return only the true elements

- **flatMap(function):**
  - Applies a function to each element of the list and flattens the lists in an element

- **reduceByKey(function):**
  - Applies a function on key-value (K,V) pairs and returns a dataset of (K, V) pairs where the values for each key are aggregated using the given reduce function, which must be of type (lambda V1,V2 : f(V1,V2)).