Apache Spark as an Ad-Hoc Query Engine for Data Science using SQL and Python

Jim Jones, Dale Legband, and RJ Smith

***Abstract*—The Spark project for MSDS 7330 Team Project for Fall Semester, 2015, Section 401. The team produced videos totaling 35 minutes, as well as the PPT deck of the information presented therein, and this report.**

***Index Terms*—Spark, Hadoop, HDFS, Python, SQL**

I. Introduction

SPARK has become a leading technology in a short period of time. Due to the recent announcements by IBM and others, we selected Spark as the topic for a video series introducing the technology. You can think of Spark as a new version – a superset – of Hadoop's MapReduce. Many challenges were encountered and solved, much learning was done, and we believe we reached our goal of producing videos that will enrich other SMU Data Sciences students' learning experience.

II. Problem Statement

A Google search for "Data Science Spark Python Education" returns 439,000 results. SMU Data Science students need to have a trusted place to begin their exploration of Spark in order to minimize wasted effort and maximize applicability with the SMU Data Science curriculum.

While exploratory learning is essential to an individual's success, so is developing the right baseline on a topic on leading edge technology in the marketplace. The typical SMU Masters of Data Science student is working a full time job, has a family, and has to balance many elements to keep up with the fast-moving SMU program. These students will appreciate having a trusted baseline readily available to them.

III. Research Methodology

First, we researched the use of Spark and learned to use Spark basics ourselves. Much of the initial effort was focused on VMs that would support our research, and we ultimately decided upon Hortonworks HDP 2.3.2 sandbox, provided for either a VirtualBox or VMware virtual machine running on Windows. This was a time saver versus DataBricks, which had to be downloaded and built on a Linux box. The HortonWorks Data Platform (HDP) requires 8GB for the VM, however all the Hadoop and Spark systems are running in the default system. This saved a lot of IT work in building and configuring a VM, which would not have lent input to the target lessons.

We produced videos with a basic introduction, and 3 drill-down videos on the following areas:

1. This first section will cover how and where to host Spark, and how to get that set up on AWS, Azure, or a standalone VM environment. Additionally, loading data from original data sources into HDFS for subsequent use in Spark will be reviewed.
2. Creation of RDDs in Python is a critical function. A basic discussion of Python as programming language for the creation of RDDs and interacting in a Spark environment is included. This section also briefly covers notebook creation in Zeppelin, an similar approach to IPython notebooks.
3. Creating and querying data in DataFrames via SparkSQL. In particular, how SQL is parsed and executed against data stored in DataFrames.

Each of these videos is a combination starting with introductory/overview slides with voice overlay and description and followed by hands-on video presentation. Each area should have 5-10 PPT slides for inclusion in final slide deck. Our demonstration use cases and data sets will be selected to highlight the functionality but not necessarily the speed or scalability due to the lack of a true cluster environment.

3.1 Introduction / Overview

Why should we care about Spark? Well, there are really two key reasons. First, Spark is going to run analytics faster on big data. That's because the complex workflows that are typically present in an analytics use case often involve iterative tasks. What Spark enables in comparison to standard Hadoop Map/Reduce is the ability to utilize and pass data that is cached in memory, which makes it much faster.

This project is submitted for meeting the database project requirements of MSDS 7330 on December 10, 2015.

Jim Jones may be reached via email at jonesjl@smu.edu. Dale Legband may be reached at dlegband@smu.edu. RJ Smith may be reached at richard@smu.edu.

.

Additionally it allows the user to be much more productive in their day-to-day tasks. Users can build their predictive models faster. Which allows them to iterate more rapidly. And because they can build multiple models concurrently without having to wait for the system, it allows them to conduct more experiments over a shorter period of time thereby enabling them to come to solutions to their analytic use cases faster than ever before.

By way of background, Spark was developed about six years ago at the AMPLab at UC Berkeley. The founders are the people who incorporated Databricks, which offers a cloud offering of Spark and commercial support for it. The technology for it was open sourced about a year later in 2010 and it became a top-level Apache project very recently in 2014.

Since 2009 Spark has had more than 800 developers from over 200 companies contribute to the project. The Open Source world allows for many people with varying degrees of skill and knowledge to contribute and improve the technology and is another key advantage of Spark.

There are multiple libraries that are included in Spark that allow programmers to use the same abstraction layer. This also makes it much more efficient because they are reusing code, reusing skills and really the goal of this technology was to generalize the old MapReduce model so that new applications can be supported within the same engine.

The Spark Core API allows for programmers to interact with it via different programming languages such as R, Python, SCALA, Java, and so on. And on top of that API, is a set of libraries that provide different kinds of capabilities. The SQL, the Streaming Technologies, Spark MLib, which is the machine learning library as well as Graph Computation with GraphX. So what this provides is a very broad set of capabilities that allow organizations to extend on top of them.

It's important to understand the relationship between Spark and Hadoop. First and foremost, Spark is not a competitor to Hadoop. Furthermore, Spark works with or without Hadoop. There are different degrees of relationship between Spark and Hadoop. On the one hand, they are very complimentary with one another. Spark by itself does not have a native storage system and therefore it integrates very well with the Hadoop and the Hadoop HDFS storage system.

Spark is a great combination with Hadoop HDFS but it can also be utilized with data that is not stored in HDFS using other kinds of storage mechanisms. So from this perspective there is a great synergy with Spark.

Spark also offers alternatives to things that are available in Hadoop. For example, Spark is able to use YARN to run on the same nodes in Hadoop as the data that's being analyzed. So it can use YARN as a resource manager and by doing this it allows the administrator to control how much resources Spark users will get versus the other workloads that are running in their Hadoop environment. So it provides this easy and discretely controlled environment to manage workload. There are also other infrastructures that Spark can run other than YARN.

The one area that Spark does compete directly with Hadoop is the area of Spark versus Map/Reduce. Spark is a competing technology to Hadoop Map/Reduce and it is an either/or selection.

3.2 Hadoop Distributions, Hosting Options, and Sandbox Implementations

Although Spark can be run in standalone mode or run with Mesos distributed processing kernel, the preferred mode is to run in conjunction with other Hadoop applications, leveraging YARN as a resource manager. Below are some of the major Hadoop distributions that support and include Spark out-of-the-box:

- Cloudera
- HortonWorks
- MapR

While each of these solutions provides a comprehensive set of Hadoop applications—Spark has been running on and included with Cloudera and HortonWorks the longest. MapR, however, includes a distinct set of advantages beyond the scope of this analysis that may make it a worthy option.

Additionally, there are several cloud providers of services that include Hadoop clusters in the cloud. Both Amazon's AWS and Microsoft's Azure provide VMs in scalable clusters of many nodes on each of the above mentioned Hadoop distributions.

In addition, there are several value-add solutions that run on top of AWS clusters that allow for more hands-off approach to cluster management, including easy spin-up of spot clusters (those that are create at time of use, then removed while not in use to save costs). The most notable managed solution is Databricks, which is a Spark-only company, whose CTO is Matei Zaharia, the creator of Spark. Databricks is a very frequent contributor to the Spark source, and always hosts and supports the most recent (as well as older) versions of Spark.

It is important to note that when provisioning in Cloud environment, that Hadoop clusters can be configured to hold data in a standard HDFS storage system, but clusters in AWS can also take advantage of S3 storage—while those in Azure can be configured to consume Azure Blob Storage, and soon Microsoft's Data Lake storage solution specifically designed for Big Data storage and analysis.

Both Cloudera and Hortonworks provide easy-to-install and use sandbox environments based on a CentOS virtual machine that runs on either VMWare or VirtualBox. While these are comparable, we chose to use the Hortonworks sandbox, as it is readily available, has relatively low overhead, and is a 100% open-source and free solution—while the other solutions are

commercial product offerings that offer much more in terms of an integrated commercially-licensed Hadoop distribution.

3.3 Using Python Spark

3.3.1 Lessons Learned

First, some of the challenges encountered in preparing the example scenarios for the videos:

- When the VM says it requires 8GB, it means 8GB. The plethora of systems that run on the Linux system is extensive.
- The purpose is to make an all-purpose sandbox for analytics, and for that, all the services have to be started. Note this is also why boot up on a decent machine takes > 3 minutes. All the services have to start. And Spark RDDs require large sums of memory.
- The INFO level of messages in the default system results in a very large number of messages when commands are successful. Sorting through the WARN messages to find significant messages is a second level of challenge.
- Elimination of the WARN messages for the default system and provided examples would be a great step forward for the Horton system.

Spark on Linux is supported by a system of software components. For example the Ranger security system and the log4j message logging system. Each system is evolving on its own with often independent philosophies and release schedules. A Computer Scientist partnering with a Data Scientist is probably an effective combination.

The response times with small data sets on a laptop are very poor. This is because the system is made to handle very large data sets, complex workloads, and run them fast. It will run the large workloads but we were not able to demonstrate scaling in this timeframe.

3.4 Key Learnings

Having the Hortonworks free Data Platform is invaluable. It shows why the open source components are being used frequently for research and for commercial purposes. Machine Learning, for example, has grown by leaps and bounds in the past 5 years to where it can now be run on HDP with Spark and the MLlib by Data Scientists.

Python is only the base language, but is extended by a plethora of add-on packages. While it has its weaknesses, this is a powerful development ecosystem that will move Data Science forward far faster than the paradigms of the past.

The concepts once started by Hadoop MapReduce are far more applicable than anyone thought in the beginning. Workloads can often run in this environment relieving SQL of the pressure and allowing it to do what it does well – structured data, with ACID requirements.

Getting into scaling and pipelining with real data sets is where we need to have this course, but that target proved out of our reach. Showing how scaled RDDs, or better scaled DataFrames, to work on a real cluster will prove invaluable to the Data Science student.

3.5 Using Python Spark SQL, Streaming, and MLlib

3.5.1 Key Challenges

Once past the basic challenges, using Spark SQL was relatively easy.

MLlib is more easily understood and demonstrated if you have a background in Machine Learning, for example the key algorithms that solve particular problems. The MLlib examples are more complex and involved than the other demonstrations.

3.5.2 Key Learnings

Learning about the DataFrames concept and incorporating those transformations was the key learning for the SQL component.

Spark + SQL + MLlib + Streaming + GraphX are a powerful combined toolkit. The goal is to create a one-stop-shop for data analytics.

The number of ML algorithms being used in the market place today is, in part, a tribute to MLlib. Every time you go to Amazon or Netflix, you are seeing ML in action.

IV. Deliverables

A. Videos

- Getting Started with Spark
  https://youtu.be/Ur6ik8r5XUg
- Setting up a Spark to Run in a VM Locally:
  https://youtu.be/sE8_sSLcAms
- First Spark Program with Python:
  https://youtu.be/aM7EFbnOwVs
- Spark for Data Science
  https://youtu.be/bJbmjfOolMc
- Other Resources:
  This document as well as the set of PowerPoint slides may be found at the following GitHub location:
  https://github.com/SMUAustinMSDS/SparkMSDS7330DatabaseProject/blob/master/README.md

V. Previous Work

Matei Zaharia's Berkeley Ph.D dissertation *An Architecture for Fast and General Data Processing on Large Clusters*, which introduced RDDs and Spark, was winner of ACM 2014 Doctoral Dissertation Award. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-12.pdf

IBM's commitment to Spark as a key technology for the future is covered in the article from InformationWeek: *IBM Bets On Apache Spark As 'The Future Of Enterprise Data'* Available: http://www.informationweek.com/big-data/ibm-bets-on-apache-spark-as-the-future-of-enterprise-data/d/d-id/1320855

Motivation for Spark is demonstrated, starting from the justification for big data sets, this video follows the logic to the motivation for Spark and where Big Data can bring an edge. Video Keynote of Strata 2014: *How Companies are Using Spark, and Where the Edge in Big Data Will Be*. Available: http://www.youtube.com/embed/KspReT2JjeE?autoplay=1

Evidence of Spark as a major tool for Data Science students to know and understand can be found in this article. eWeek Article *Apache Spark Continues to Gain Enterprise Traction*. Available: http://www.eweek.com/database/apache-spark-continues-to-gain-enterprise-traction.html

Motivation for Spark when MapReduce already exists is covered in this article: *MapReduce and Spark.* Available: http://vision.cloudera.com/mapreduce-spark/

Examples of current training material are saved at the Spark Summit 2014 Training Archive. Available: https://spark-summit.org/2014/training

VI. Resources

A. Reference Websites and Books

[1] Apache Spark Official Website. Available: http://spark.apache.org/

[2] Examples of Spark Programming in Python. Available: https://spark.apache.org/examples.html

[3] Hortonworks Official Website. Available: http://hortonworks.com/

[4] Spark Wiki. Available: https://cwiki.apache.org/confluence/display/SPARK/Wiki+Homepage

[5] Databricks Spark Page. Available: https://databricks.com/spark/about

[6] Apache Spark YouTube Channel. Available: http://www.youtube.com/channel/UCRzsq7k4-kT-h3TDUBQ82-w

[7] Wes McKinney. (2012, Nov. 1) *Python for Data Analysis* (1st edition) (O'Reilly Media)

Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills. (2015, April 20) *Advanced Analytics with Spark.* (1st edition) (O'Reilly Media)

[8] Holden Karau, Andy Konwinski, Patrick Wendell and Matei Zaharia. (2015, May 8) *Learning Spark.* (3rd release) (O'Reilly Media)

[9] Nick Pentreath. (2015, Feb. 20) *Machine Learning with Spark* (Packt Publishing)

[10] Rishi Yadav. (2015, July 27) *Spark Cookbook*. (Packt Publishing)

B. Spark Hosting Environment

To facilitate the research, we applied for "grant" resources through Databricks Academic Partners Program and through Amazon's AWS Education gateway program.

Our AWS request was granted, but only in the last week, so we will not be provisioning through AWS virtual machine cluster.

C. Datasets

As our starting point for developing examples, we have used www.apache.org and www.hortonworks.com base material and their pointers to simple example data sets that are installed by default. While the data sets used were very small, the functions were demonstrated.

D. Video Recording and Playback

Our investigation into video recording software and selection was not productive. Each company that we selected for providing software oriented to making online courses - eLearning solutions - have free evaluations only for faculty, not students.

Our search was not extensive or comprehensive as we decided the technology was less important than the content. Therefore, we went to SnagIt, which has basic video capture capabilities and recorded with that package. Limits include not having a webcam view of our faces and clumsy interactions between PowerPoint slides and switching to using the software. Video playback will be hosted on YouTube.