CSC Independent Study (CSC 630) - Non thesis Masters

Fall 2015

Final Report

Student Name	Saurabh Vishwas Joshi
ID	200060154 (sjoshi6)
Instructor	Dr. Christopher Parnin
Project Title	AutoSpark
Credits	3
Date	December 1, 2015

AutoSpark

A tool to auto configure Apache Spark clusters for text analysis

Description:

AutoSpark is a tool to automatically configure Apache Spark clusters in stand alone mode on Amazon EC2 and DigitalOcean. It enables the data analysts to quickly leverage the power of Apache Spark for text analysis. Apache Spark processes and stores data in memory. Thus, the nodes used in an Apache Spark cluster are extremely high performant and incur a high cost. To keep the operational costs low it is essential that Apache Spark clusters be started and shut down on a frequent on-demand basis. Repeated manual configuration of Spark clusters is tedious and involves a significant amount of configuration. This renders the manual solution to be error prone and time consuming. Automating the setup process for Spark clusters helps resolve this issue. This is precisely the goal of the AutoSpark tool. With a few commands, a user can quickly start an Apache Spark cluster of the desired size, submit a job remotely, and eventually shut it down. The tool abstracts away all the setup and configuration complexities from the user.

Technology Stack:

Node JS	- for spinning up a Spark Cluster on Digital Ocean
Python	- for spinning up a Spark Cluster on Amazon AWS
Apache Spark	- The Spark framework for big data analysis
DigitalOcean / Amazon AWS	- Cloud Providers
Shell Scripts	- Backend Drivers

Outcomes

The AutoSpark tool is completely functional and is verified to be working with Amazon AWS and Digital Ocean on Ubuntu OS. The tool supports below mentioned functionalities:

- Launching of Apache Spark Cluster in Standalone mode
- Allows the user to spin up a new cluster using the command line driver
- Allows the user to load the data into the cluster
- Allows the user to submit a PySpark Job remotely to the Apache Spark Cluster
- Allows the user to tear down the cluster.
- A docker image file to automatically use the tool without initial machine setup.
- The tool has been tested with a simple log scanner program that detects success rate of an Apache Web server using log processing.
- Experimentation has also been done on distributing JSON based data files across the cluster.

Findings

- Apart from the Standalone mode Spark clusters can also be setup in a Mesos / HDFS compatible mode.
- Distribution of data on Spark clusters is time consuming. Instead, using an HDFS supported data source such as S3/ HDFS / Cassandra is a more preferred methodology.
- Setting up the AutoSpark tool requires certain dependancies to be present on a machine. This has been abstracted out by the use of Docker Containers.
- Amazon AWS access keys and Digital Ocean API Tokens are extremely sensitive. Users end up uploading these credentials along with the code. The application removes this issue by prompting the user to insert these credentials at runtime instead of storing them.

Conclusion

The AutoSpark tool is an efficient and time saving solution for using Apache Spark clusters on Cloud vendors. It is the most ideal for users who intend to frequently recycle their Spark clusters. It removes the manual errors from the configuration process and supports remote job submission to the cluster.

Further Improvements

- On boarding more cloud vendors to the AutoSpark tool (Microsoft Azure).
- Supporting cluster launch in modes other than the standalone mode.
- Creating a compiled version of the application.
- Uploading a docker image on docker hub.
- Creating a sample PySpark application that uses data stored in Amazon S3.

Project Link:

- https://github.com/alt-code/AutoSpark.git