Pranav Kavikondala pk6994

Atreya Misra am73676

Sriram Ravula sr39533

Term Project: Spark Word Count

***Appendix A: Introduction to Hadoop and Spark***

The Apache Spark implementation first requires one to have Hadoop Yarn running. To set up Hadoop on a mac machine, one can install Hadoop using Homebrew, a package manager for macOS. Following the instructions given in Assignment 5, one will be able to successfully change the configuration files of Hadoop to configure the implementation with Hadoop Yarn. Once Yarn is configured, download the latest version of Spark (2.1.0), unzip the tar.gz archive, then Spark will be ready to run on a mac machine [1]. Now that Spark is set up, one must understand the important differences between Hadoop and Spark.

Hadoop uses something called the Hadoop Distributed File System (HDFS). HDFS is a distributed file system that allows storing files across multiple machines, and comes in handy for very large file processing (think terabytes or larger) [2]. The most important tool when writing a Java program to run on the HDFS is MapReduce. MapReduce takes in key value pairs from an input file as inputs to the mapper [2]. One does not define the key value pairs, only the variable types. Then, the implementation is up to the programmer for the mapper and the reducer. The mapper takes in the key value pairs and transforms them based on the implementation sought by the programmer. However, due to the HDFS, these transformations are separated. Hence, the programmer also implements reduce, which aggregates all the outputs of the mapper into an interpretable output. In the example of word count, the mapper will concurrently find word counts of different lines based on the partition of how the HDFS feeds the lines of different inputs. Next, the overall word count will be found by aggregating all those values together using reduce.

Another important aspect of Hadoop is YARN. YARN controls the resource management of Hadoop applications and scheduling of applications on different domains (on a machine by machine basis) [3]. An important aspect of Spark is that it can bind onto Hadoop YARN to run on the Hadoop Cluster. Spark takes advantage of the Hadoop cluster in a different way than Hadoop. Instead of the Hadoop implementation of HDFS, which writes to a disk in intermediate steps, Spark processes the data in memory after copying it from the HDFS server [4]. Running on RAM allows Spark to cut its time interacting with servers and run a lot faster than a similar Hadoop MapReduce implementation would. However, because Spark is using RAM, there is the possibility of having too much data that may not fit in memory. In turn, a Spark application could more easily crash or give corrupt outputs compared to the HDFS. To avoid crashing, Spark uses a Resilient Distributed Dataset (RDD). RDDs work by creating restricted form of shared distributed memory to avoid different processes accessing the same memory and allows for recovery of data when there is a failure [4]. When running Spark, one must understand the limitations of the distributed system to avoid exceeding them, even though RDDs implement fault tolerance. It may be more practical to use Hadoop’s implementation of HDFS if the system is not optimized to run Spark.

References

[1]“Running Spark on YARN,” Running Spark on YARN - Spark 2.1.0 Documentation. [Online]. Available: http://spark.apache.org/docs/latest/running-on-yarn.html. [Accessed: 18-Apr-2017].

[2]“HDFS Architecture Guide,” HDFS Architecture Guide. [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs\_design.html. [Accessed: 18-Apr-2017].

[3]“Apache Hadoop YARN,” Apache Hadoop 3.0.0-alpha2 – Apache Hadoop YARN. [Online]. Available: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html. [Accessed: 18-Apr-2017].

[4]M. Jeevan, “Spark vs. Hadoop - All You Need To Know,” Simplilearn.com, 25-Oct-2016. [Online]. Available: https://www.simplilearn.com/spark-vs-hadoop-article. [Accessed: 18-Apr-2017].