Title: Large-scale Workload Characterization in Apache Spark Framework.

Co-authors: Debzani Deb, Winston-salem state university

Recently, Apache Spark platform [1] has grown both in popularity and complexity and has been increasingly adopted by a wide spectrum of areas for big data analytics and many of them use cloud environments to deploy their applications. In order to use available cloud resources in an efficient way and at the same time to effectively meet the workloads demands, spark user need to understand how the execution of her application is impacted by the various configuration settings. Spark relies on users for specifying many configuration parameters and understanding the affect of these choices with respect to the execution time and resource utilization is not an easy task. The goal of this study is to investigate the characteristic of spark application performance through running them with various configuration settings and to gain deep insight about the resource provisioning, workload management etc. Toward this goal, we implemented a Spark on YARN cluster on the Chameleon Cloud [2], a large-scale and open cloud research platform funded by the NSF that allows for greater computational power that we lack having in house. Our cluster has master and 8 slave nodes (each with 48 cores, 128 GB memory) running Ubuntu 16.04 OS with Hadoop 2.7.4, Spark 2.2.0, and Scala 2.13.8. We gathered our experimental data based on the logged information available through Spark's history server and by utilizing cluster-wide monitoring tool Ganglia, that provides us important insight about the overall cluster resource utilization (such as CPU and memory). We used two representative Spark-SQL applications from TPC-H [3] benchmark and executed them with 50GB and 100GB datasets. We choose Query 1 (Q1) and Query 5 (Q5) because of their unique characteristics. Q1 has minimal join operation (therefore least shuffling data), which makes it to be CPU bound, on the other hand, Q5 is an I/O bound job that contains multiple join operations (significant shuffling during stages). Our results shows that, the two different applications exhibits different resource consumptions and their execution times are significantly impacted by the settings of the configuration parameters such as size of the RDD cache, the number of concurrent tasks in each executor, executor configuration (core and memory per executor) etc. Configuring a remote Spark on Yarn cluster from scratch with all the abovementioned tools is a significantly challenging task and we learned a lot throughout the process. We are really excited to present our learning experiences and our experimental results during ERN 2018 conference and hope to receive important feedback that will further us with our ultimate research goal of investigating Elastic Spark where resource provisioning for spark executors in multi-tenant environment can occur dynamically and informatively.

References

1. Apache Spark, https://spark.apache.org
2. Chameleon cloud, https://www.chameleoncloud.org
3. TPC-H is a Decision Support Benchmark, http://www.tpc.org/tpch/

Fund to be acknowledged**:** This study is supported by NSF HBCU-UP grant #1600864 awarded to Debzani Deb, Associate Professor, Winston-salem state university.