SPARK Job Performance Analysis and Prediction Tool(DEMO)

Rekha Singhal
TCS Research
India
rekha.singhal@tcs.com

Chetan Phalak
TCS Research
India
chetan1.phalak@tcs.com

Praveen Kumar*
TCS Research
India
praveenhelp78@gmail.com

ABSTRACT

Spark is one of most widely deployed in-memory big data technology for parallel data processing across cluster of machines. The availability of these big data platforms on commodity machines has raised the challenge of assuring performance of applications with increase in data size. We have build a tool to assist application developer and tester to estimate an application execution time for larger data size before deployment. Conversely, the tool may also be used to estimate the competent cluster size for desired application performance in production environment. The tool can be used for detailed profiling of Spark job, post execution, to understand performance bottleneck. This tool incorporates different configurations of Spark cluster to estimate application performance. Therefore, it can also be used with optimization techniques to get tuned value of Spark parameters for an optimal performance. The tool's key innovations are support for different configurations of Spark platform for performance prediction and simulator to estimate Spark stage execution time which includes task execution variability due to HDFS, data skew and cluster nodes heterogeneity. The tool using model [3] has been shown to predict within 20% error bound for Wordcount, Terasort,Kmeans and few SQL workloads.

CCS CONCEPTS

• **General and reference** → *Performance*;

KEYWORDS

Spark, Performance, Prediction

ACM Reference Format:
Rekha Singhal, Chetan Phalak, and Praveen Kumar. 2018. SPARK Job Performance Analysis and Prediction Tool(DEMO). In *Proceedings of ACM/SPEC International Conference on Performance Engineering (ICPE'18 Companion).* ACM, New York, NY, USA, 2 pages. https://doi.org/http://dx.doi.org/10.1145/3185768.3185772

1 INTRODUCTION

Apache Spark is one of the widely deployed commodity cluster big data platforms available in open source for in-memory parallel processing. Application deployment on commodity Spark cluster system has challenge of assuring its performance over time with increase in data size. Conversely, appropriate capacity sizing of production system is needed for desired performance irrespective of increase in data size. This raises the need for a performance assurance model and tool, which can estimate an application performance for larger data sizes and variable cluster sizes before deployment. Here, by performance we mean application *execution time*.

We had built an analytic model to predict a Spark job execution time for larger data size using its measurements on small data size in small Spark cluster [3]. One of the key component of the model is Spark's stage simulator to estimate a stage execution time. Similar approach has been used for MR job performance prediction as well [4]. The model can be used for different Spark configurations by varying Spark parameters which can be changed during application execution and hence the tool may be used with optimization techniques to get tuned value of Spark parameters for auto tuning. [1, 2] discuss instrumentation of Spark job for tuning, however, we employ grey box approach. The paper is organized as follows. Section 2 discuss the performance prediction model [3] in brief. Section 3 presents the tool features and architecture in detail and finally the paper is concluded in Section 4.

2 SPARK APPLICATION PERFORMANCE MODEL

We assume a small size Spark cluster with application and its representative data sets available in development environment. The cluster is assumed to have atleast one instance of each type of heterogeneous nodes deployed in production system. The application is executed in this small cluster on small data size (*DevSize*). An application is executed as a serial execution of a number of Spark jobs.A Spark job is executed in a form of directed acyclic graph (DAG), where each node in the graph represents a stage. A new stage is created whenever next operation requires data to be shuffled. A job's execution time is predicted as summation of the estimated execution time of all its stages. Each stage is executed as set of concurrent executors with parallel tasks in each executor, depending on values of *number of executors* and *number of cores per executor* parameters respectively. Because of hetereogenity of nodes, HDFS and data skewness the symmetry of task execution across different core is broken, and we have built a stage simulator to estimate a stage execution time for larger number of tasks or data size.

The application logs created by Spark platform are parsed to collect granular level performance data as given in [3]. A stage tasks are divided into two waves - first wave and rest wave tasks. The granular level measurements are used to estimate task's scheduler delay, task's JVM time and task's shuffle time for both the waves

*Praveen has contributed to this work while being at TCS Research

ICPE'18 Companion, April 9–13, 2018, Berlin, Germany

ACM ISBN ISBN 978-1-4503-5629-9/18/04…$15.00
https://doi.org/http://dx.doi.org/10.1145/3185768.3185772

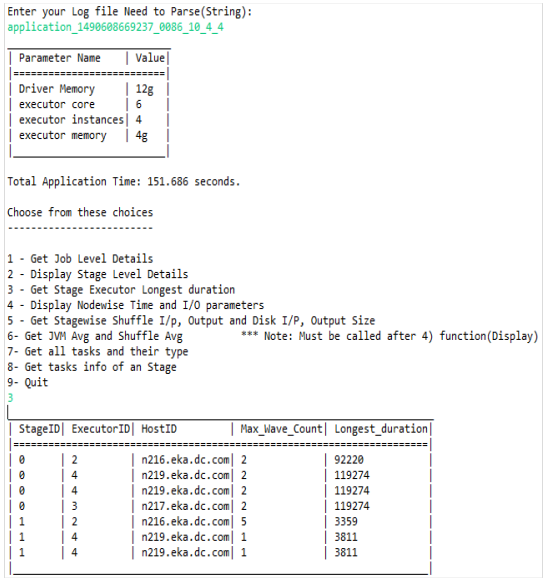

Figure 1: Spark Job Analyzer Screen

for larger data and cluster size. A stage execution is simulated for larger data size using the task's estimated execution time which gives stage's estimated execution time.

3 TOOL ARCHITECTURE

The tool has two components - Spark job Log Analyzer (SLA) and Spark job Performance Predictor (SPP). SLA parses and analyzes a Spark job log after its execution. SPP uses model [3] to predict a Spark job execution time.

3.1 Spark Job Log Analyzer

Spark Log Analyzer(SLA) parses a Spark job log after its execution and collects all stage and task level time details. SLA processes granular level data to display meaningful higher level performance counters. This module has challenge of using data structures efficiently to process large data sets. SLA outputs number of waves, set of tasks started and finished in parallel in almost same time window, in a job execution. The tool also displays data for disk read and writes, clean up time and split of execution time at various granular levels such as per stage, per task, per executor and per node. Once a Spark job log file with its absolute path is provided, SLA displays four parameters on screen as shown in Fig 1. Below that, eight options are provided to an user for selection. User can select option as per requirement and view the specific data.

3.2 Spark Job Performance Prediction Tool

The tool is comprised of three components Log Parser, Model Builder and Spark Job Execution Time Predictor as shown in Fig 2.

- Log Parser : Input to the Log Parser is the log file generated on execution of a Spark job on small data and cluster size. This is constrained version of SLA. It parses the execution log file, required and filtered contents are further provided to the job execution time predictor.

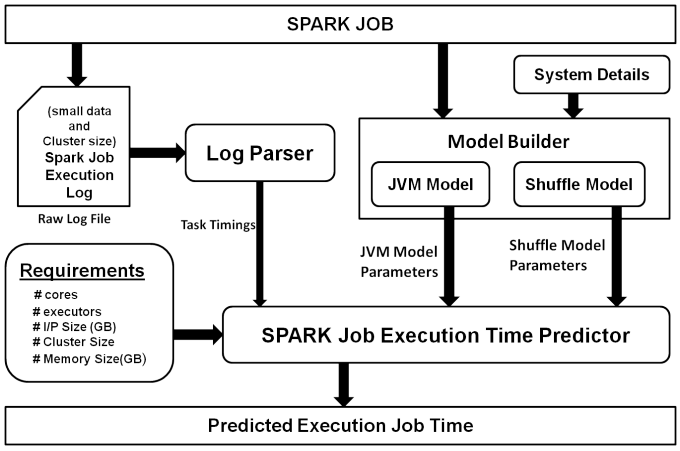

Figure 2: Spark Job Performance Estimation Tool Architecture

- Model Builder: Model builder uses system configuration details such as spark memory size and number of cores per node, size of cluster and network bandwidth to simulate Spark job execution on different configurations to build JVM and Shuffle time prediction models. These models' parameters are stored in plain text files and used by Spark job execution time predictor.
- Spark Job Execution Time Predictor: This uses measurements given by Parser, JVM and Shuffle models to predict a given Spark job execution time for the production configuration and data size given in 'Requirements' file as shown in Fig 2, using model [3].

4 CONCLUSIONS

The wide availability of commodity based big data platforms has raised the challenge of assuring application performance post deployment with increase in data size. In this paper, we have presented a tool which can be used to estimate a Spark job execution time for larger data sizes, using measurements of the job execution on small data size in small cluster, before deployment. The tool is based on the model presented in [3] and can predict with an average accuracy of 80%. The tool can also be used to do detailed analysis of the Spark job execution to understand the performance bottlenecks.

REFERENCES

[1] Michael Armbrust, Tathagata Das, Aaron Davidson, Ali Ghodsi, Andrew Or, Ion Stoica Josh Rosen, Patrick Wendell, Reynold Xin, and Matei Zaharia. 2015. Scaling Spark in the Real World Performance and Usability. In *Proceedings of the VLDB Endowment 41st International Conference on Very Large Data Bases.*

[2] Kay Ousterhout, Ryan Rasti, Sylvia Ratnasamy, Scott Shenker, and Byung Chun. 2015. Making Sense of Performance in Data Analytics Frameworks. In *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI).*

[3] Rekha Singhal and Praveen Singh. 2017. Perfomance assurance Model for Applications on Spark Platform. In *In Proceedings of TPCTC held in conjunction with VLDB, Springer Veralag.*

[4] Rekha Singhal and Abhishek Verma. 2016. Predicting Job Completion Time in Heterogeneous MapReduce Environments. In *Proceedings of IPDPS: Heterogeneous computing workshop, IPDPS.*