Job Performance Overview of Apache Flink and Apache Spark Applications

Jan Frenzel
René Jäkel
jan.frenzel@tu-dresden.de
rene.jaekel@tu-dresden.de
Technische Universität Dresden
Dresden, Germany

KEYWORDS

big data,performance analysis,sampling,trace,Apache Spark,Apache Flink

ACM Reference Format:
Jan Frenzel and René Jäkel. 2019. Job Performance Overview of Apache Flink and Apache Spark Applications. In *Proceedings of The International Conference for High Performance Computing, Networking, Storage and Analysis (SC19).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Apache Spark and Apache Flink are two Big Data frameworks used for fast data exploration and analysis. Both frameworks provide the runtime of program sections and performance metrics, such as the number of bytes read or written, via an integrated dashboard. Performance metrics available in the dashboard lack timely information and are only shown aggregated in a separate part of the dashboard. However, performance investigations and optimizations would benefit from an integrated view with detailed performance metric events.

2 PROPOSED APPROACH

We propose a system that samples metrics at runtime and collects information about the program sections after the execution finishes. The program sections and their start and end times are provided by the history server, a runtime component that stores information about finished application runs. For the metrics, we enable the built-in monitoring interface of the frameworks. Then, probes are inserted into the framework processes to collect framework- and JavaVM-related performance metrics via the Java management extensions (JMX). The probes sample user-defined metrics in regular time intervals. For each sample, the metric value is stored together with a timestamps in a trace file formatted in Open Trace Format 2 (OTF2). When a Spark or Flink application finishes, start and end times of its program sections are obtained from the history server and converted to OTF2. Trace files with exact time information and values of metrics are combined with timely information of program sections and can be displayed and analyzed with performance tools such as Vampir[2].

We ran KMeans, a parallel clustering application, from the HiBench benchmark suite[1] on a six node cluster to validate our approach.

3 SUMMARY

The performance data is stored in an established format independent from Spark and Flink versions and can be viewed with state-of-the-art performance tools, i. e. Vampir. The overhead depends on the sampling interval and was below 10% in our experiments. The system allows to enrich the traces with further data in the future, such as call stack data gathered via Java bytecode instrumentation.

4 ACKNOWLEDGMENTS

This work was supported by the German Federal Ministry of Education and Research (BMBF, 01IS18026) by funding the competence center for Big Data "ScaDS Dresden/Leipzig".

REFERENCES

[1] Shengsheng Huang, Jie Huang, Jinquan Dai, Tao Xie, and Bo Huang. 2010. The HiBench benchmark suite: Characterization of the MapReduce-based data analysis. In *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010).* IEEE, 41–51.

[2] Andreas Knüpfer, Holger Brunst, Jens Doleschal, Matthias Jurenz, Matthias Lieber, Holger Mickler, Matthias S. Müller, and Wolfgang E. Nagel. 2008. The Vampir Performance Analysis Tool-Set. In *Tools for High Performance Computing*, Michael Resch, Rainer Keller, Valentin Himmler, Bettina Krammer, and Alexander Schulz (Eds.). Springer Berlin Heidelberg, 139–155. https://doi.org/10.1007/978-3-540-68564-7_9

SC19, November 17–22, 2019, Denver, CO

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn