Benchmark Stream Processing Systems

Yangjun Wang

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 28.12.2015

Thesis supervisor:

Assoc. Prof. Aristides Gionis

Thesis advisor:

D.Sc. Gianmarco De Francisci Morales

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Yangjun Wang

Title: Benchmark Stream Processing Systems

Date: 28.12.2015 Language: English Number of pages: 5+9

Department of Information and Computer Science

Professorship: Data Communication Software

Supervisor: Assoc. Prof. Aristides Gionis

Advisor: D.Sc. Gianmarco De Francisci Morales

Batch processing technologies(Such as MapReduce, Hive, Pig) have matured and been widely used in the industry. These systems solved the issue processing big volumes of data successfully. However, first big data need to be collected and stored in a database or file system. Then it takes time to finish batch processing analysis job before get any results. While there are many cases that need analysed results from streaming data immediately. The demand for processing real time stream data is increasing a lot these days. A big data architecture contains several parts. Often, masses of structured and semi-structured historical data are stored in Hadoop (Volume + Variety). On the other side, stream processing is used for fast data requirements (Velocity + Variety)[1]. Several streaming processing systems are implemented and widely adopted, such as Apache Storm, JStorm, Apache Spark, IBM InfoSphere Streams and Apache Flink. They all support real-time stream processing, high scalability, and awesome monitoring. How to evaluate a real time stream processing system before choosing it to use in production development is a open question. Before these real time stream processing systems are implemented, Michael demonstrated the 8 requirements[2] of real-time stream processing, which gives us a standard to evaluate whether a real time stream processing system satisfies these requirements. A very common and traditional approach to verify whether the performance of a system meets the requirements is benchmarking. Published benchmarking results from industry standard benchmark systems could help users compare products and understand features of a system easily.

Keywords: Big Data, Stream, Benchmark, Storm, Flink, Spark

Acknowledgements

I want to thank ProfessorAristides Gionis and my advisor Gianmarco De Francisci Morales for their good guidance.

Otaniemi, 16.1.2015

Yangjun Wang

Contents

1 Introduction

Introduce big data and the four ***V***s of Big Data

1.1 Stream Processing Systems and Evaluation

Describe common stream processing systems and current evaluation and comparison these platforms.

1.2 Structure of the Thesis

Structure description of the thesis

2 Background

Background knowledge of StreamBench which includes Big Data, Cloud Computing and widely accepted benchmark systems.

2.1 Cloud Computing

Concept of Cloud Computing and how Cloud Computing solves Big Data issues

2.1.1 Parallel Computing

2.1.2 Computing Cluster

2.1.3 Batch Processing and Stream Processing

2.2 Apache Hadoop

Introduce Apache Hadoop and several important modules

2.2.1 MapReduce

2.2.2 Hadoop Distribution File Systems

2.2.3 YARN

2.2.4 Zookeeper

2.3 Benchmark

Describe benchmark systems of traditional database and cloud service systems. Demonstrate design and components of benchmark system

2.3.1 Traditional Database Benchmark

2.3.2 Cloud Service Benchmark

3 Stream Processing Platforms

Introduce three widely used stream processing platforms, point out core concepts and key features

3.1 Apache Storm

3.1.1 Storm Architecture

3.1.2 Computing Model

3.2 Apache Flink

3.2.1 Flink Architecture

3.2.2 Memory Management

3.2.3 Flink Streaming

3.3 Apache Spark

3.3.1 Resilient Distributed Datasets(RDDs)

3.3.2 Spark Streaming

4 Benchmark Design

4.1 Architecture

4.2 Experiment Environment Setup

4.3 Data Source

4.3.1 Test Data Generation

4.3.2 Kafka

4.4 Experiment Log and Statistic

4.5 Extensibility

5 Experiment

5.1 Experiment Environment

5.2 Classic Workload

5.2.1 WordCount

5.2.2 Data Source

5.2.3 Algorithm Description

5.2.4 Results and Discussion

5.3 Multi-Streams Join Workload

5.3.1 Advertisements Click

5.3.2 Data Source

5.3.3 Algorithm Description

5.3.4 Results and Discussion

5.4 Iterate Workload

5.4.1 WordCount

5.4.2 Data Source

5.4.3 Algorithm Description

5.4.4 Results and Discussion

6 Conclusions

Summary of experiment results

6.1 Selection in Practice

Summarize several factors which affect selection of stream processing systems in practic

6.1.1 Performance Summary

6.1.2 Issues

6.2 Future Work

Future works

6.2.1 Scale-out and Elasticity Evaluation

6.2.2 Evaluation of Other Platforms

References

[1] K. Wer. (2014) Real-time stream processing as game changer in a big data world with hadoop and data warehouse. [Online]. Available: http://www.infoq.com/articles/stream-processing-hadoop

[2] M. Stonebraker, U. Çetintemel, and S. Zdonik, "The 8 requirements of real-time stream processing," *ACM SIGMOD Record*, vol. 34, no. 4, pp. 42–47, 2005.

A Source Code

A.1 WordCount

A.2 Advertisements Click