CE 6304 – 001 Computer Architecture, Spring 2016

PROJECT PROPSAL

A case study of parallel architecture MapReduce and Hadoop

Yifan Xu yxx152530

Yu Zheng yxz153031

Huixiang Ye hxy153530

Zhenqiao He zxh151630

1. Summary

Nowadays business world and researchers increasingly rely on computer analysis of large volume of data to make decisions or collect research evidence [1]. However, high velocity of data is created in different format every day. Hence, it is a significant challenge to improve computer performance on storing and processing large amount of data. The aim of the project is to investigate in parallel architecture and programming, which is one of the most important method to enhance computer performance [2]. Concretely, the project will evaluate the impact of MapReduce [5], which is one of the prestigious data-level parallelism model, and one of its application Hadoop on improving the computer performance [6].

2. Problems

According to IBM: "Every day, 2.5 billion gigabytes of high-velocity data are created in a variety of forms, such as social media posts, information gathered in sensors and medical devices, videos and transaction records." Data sets are growing rapidly in part because they are increasingly gathered by cheap and numerous information-sensing mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers and wireless sensor networks. These data sets require high performance computing technology to exploit valuable information that has significant potential to improve business process and facilitate scientific research. [3] Thus, this project intends to evaluate how parallel computer architecture such as MapReduce improve computer performance.

Traditionally, industry and academia use Relational Database Management Systems (RDBMS) and desktop statistics and visualization packages to store and process data set. However, these methods have limitation in meeting big data requirement in two major aspects. First, RDBMS has highly strict requirement on data structure. Since nowadays data are created in various format, it is inefficient to spend a large amount of computing power into managing data format [4]. Second, RDBMS has less scalability compare to distribute parallel computing model like MapReduce [4]. This lack of competency limits the RDBMS and similar computer architecture on keeping pace with rapidly scaling data set. On the other hand, MapReduce provides distributed processing schema to handle large volumes of structured and unstructured data more efficiently than the traditional enterprise data warehouse [7][8].

3. Objectives

Our goal of this project is to evaluate the impact of parallel architecture MapReduce and its implementation Hadoop on computer performance when processing large data set.

Specifically, we will develop a program computing word co-occurrence matrices on Hadoop platform. Then deploy the program on different computer schemas (including AWS cloud

platform with one and two instances, one personal desktop computer). Also, we will develop another program without parallel computer scheme running on the same task as a controlled group. We will record the performance of each scenario and analyze performance data.

4. Procedures

This project is divided into 8 steps, which are:

1. Analyze program requirement and design algorithm with and without MapReduce model.
2. Prepare raw data for processing.
3. Setup Hadoop development environment on different platforms mentioned above.
4. Develop programs on Hadoop platform and on non-Hadoop schema.
5. Store the data on Hadoop Distributed File System (HDFS).
6. Deploy the program on different scenario.
7. Run program and record the performance data.
8. Analyze the performance data.

5. Research Methodology

5.1 MapReduce

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. Conceptually similar approaches have been very well known since 1995 with the Message Passing Interface standard having reduce and scatter operations.

A MapReduce program is composed of a Map() procedure (method) that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a Reduce() method that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). The "MapReduce System" (also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance.

5.2 Apache Hadoop

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework.

Hadoop consists of two major components -- Hadoop Distributed File System and Hadoop MapReduce.

Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;

Hadoop MapReduce – an implementation of the MapReduce programming model for large scale data processing.

5.3 Amazon Web Services

Amazon Web Services (AWS), is a collection of cloud computing services, also called web services, that make up a cloud-computing platform offered by Amazon.com. These services operate from 12 geographical regions across the world. The most central and well-known of these services arguably include Amazon Elastic Compute Cloud, also known as "EC2", and Amazon Simple Storage Service, also known as "S3". Amazon markets AWS as a service to provide large computing capacity more quickly and more cheaply than a client company building an actual physical server farm.

5.4 Research Methodology

Our main objective is to evaluate the impact of MapReduce model and its implementation Hadoop on computer performance of processing large data set. To evaluate their performance, we will exanimate the run time on building large word co-occurrence matrices, a simple task that underlies many NLP algorithms. Further, we will run the program on Hadoop platform with different number of instances in order to evaluate the scalability of this model, which is a key matric to evaluate different model on big data processing performance. We will mainly run our conduct our experiment on amazon cloud computing service, relieving us from constructing expensive computing cluster on ourselves, in which case we can run our experiment within fifty dollars. Also AWS will generate all computer performance statistics for project analysis.

Reference

[1] Lohr S. The age of big data[J]. New York Times, 2012, 11.

[2] Hennessy J L, Patterson D A. Computer architecture: a quantitative approach[M]. Elsevier, 2011.

[3] McClean A, Conceicao R, O'halloran M. A comparison of MapReduce and parallel database management systems[C]//ICONS 2013 The Eighth International Conference on Systems. 2013: 64-68.

[4] Lee K H, Lee Y J, Choi H, et al. Parallel data processing with MapReduce: a survey[J]. AcM sIGMoD Record, 2012, 40(4): 11-20.

[5] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[6] Hadoop A. Hadoop[J]. 2009-03-06]. http://hadoop. apache. org, 2009.

[7] McClean A, Conceicao R, O'halloran M. A comparison of MapReduce and parallel database management systems[C]//ICONS 2013 The Eighth International Conference on Systems. 2013: 64-68.

[8] Lin J. Scalable language processing algorithms for the masses: A case study in computing word co-occurrence matrices with MapReduce[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008: 419-428.