Chapter 4
Beyond Hadoop:
Recent Directions in Data Computing for Internet Services

Zhiwei Xu
Chinese Academy of Sciences, China

Bo Yan
Chinese Academy of Sciences, China

Yongqiang Zou
Tencent Research, China

ABSTRACT

As a main subfield of cloud computing applications, internet services require large-scale data computing. Their workloads can be divided into two classes: customer-facing query-processing interactive tasks that serve hundreds of millions of users within a short response time and backend data analysis batch tasks that involve petabytes of data. Hadoop, an open source software suite, is used by many Internet services as the main data computing platform. Hadoop is also used by academia as a research platform and an optimization target. This paper presents five research directions for optimizing Hadoop; improving performance, utilization, power efficiency, availability, and different consistency constraints. The survey covers both backend analysis and customer-facing workloads. A total of 15 innovative techniques and systems are analyzed and compared, focusing on main research issues, innovative techniques, and optimized results.

1. INTRODUCTION

Large scale data computing is a main technology driver for the many Internet services we see today. There data computing manifests as two classes of workloads. The first are usually called *customer-facing* applications that process many interactive requests from hundreds of millions of users within a short response time. The second are called *backend* applications that conduct data analysis jobs in batch mode, where each job may involve petabytes of data.

DOI: 10.4018/978-1-4666-1879-4.ch004

This paper reviews recent research directions in data computing for Internet services. We identify five inter-related research directions that are interesting to any cloud system with data computing workloads: (1) improving performance (response time, throughput, or job execution time); (2) improving system utilization (CPU, disk, and I/O bandwidth, etc.); (3) improving energy efficiency (saving power or energy); (4) improving system availability (availability and reliability); and (5) considering different consistency constraints. The last direction is important not only because consistency affects performance, but also due to the CAP theorem (Brewer, 2000): consistency constraints affect availability and partitioned fault tolerance.

Many Internet services now use Hadoop as their main data computing platform (Hadoop, n.d.). Hadoop is also extensively used by academia as a research platform and an optimization target. Supported by a vibrant community, with increasing contributions from companies and academia, Hadoop is developing rapidly. For instance, Hadoop was originally created to handle backend data analysis jobs, including only MapReduce, HDFS and a core. Components in the current Hadoop suite (Figure 1), except HBase, are still mainly for backend applications. However, much research work is ongoing to extend Hadoop for customer-facing applications. Since Hadoop is an open source software suite organized by Apache Software Foundation, research contributions are not hindered by proprietary barriers. This paper focuses on researches that have been or can be converted and integrated into Hadoop.

The rest of the paper is organized as follows: Section 2 discusses optimization techniques for backend data analysis workloads, to improve job execution time, system utilization and energy efficiency, through innovative techniques in scheduling, I/O organization and nodes disabling. Section 3 discusses optimization techniques for customer-facing interactive workloads. We survey five systems with different data models and review three optimization techniques for specific data model operations. Section 4 offers concluding remarks and points out future research problems. We only select techniques and systems that are representative. A total of 15 innovative techniques and systems are analyzed and compared, focusing on their main research issues, innovative techniques, and optimized results.

2. OPTIMIZATIONS ON BACKEND DATA ANALYSIS

This section analyzes seven techniques for optimizing data computing for backend data analysis workloads. The objectives are to improve job execution time, system utilization and energy efficiency, involving innovative scheduling, I/O organization, node disabling techniques. Section 2.1 introduces three Hadoop scheduling optimization techniques. Section 2.2 discusses three enhancements using storage techniques. Section 2.3 reviews a technique to improve energy efficiency. Section 2.4 summarizes these techniques.

Figure 1. Hadoop structure and subprojects (Hadoop, n.d.)