Advances in Engineering Software 95 (2016) 28–37

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

ARLS: A MapReduce-based output analysis tool for large-scale simulations

Kangsun Lee*, Kwanghoon Jung, Joonho Park, Dongseop Kwon

Department of Computer Engineering, Myongji University, MyongJiRo 116, Yongin, Kyunggi-Do 449-729, South Korea

ARTICLE INFO

Article history:
Received 2 July 2015
Revised 28 January 2016
Accepted 31 January 2016
Available online 22 February 2016

Keywords:
Large-scale simulation
Simulation-based analysis
Distributed computing
Cloud computing
Hadoop and MapReduce
Simulation output analysis

ABSTRACT

As simulations are becoming popular in the analysis of the complex behavior of large-scale systems with immense inputs and outputs, there is an increasing demand to efficiently store, manage, and analyze massive simulation outputs. Hadoop and MapReduce have been used in various applications to speed up the process of analyzing large amounts of datasets. In this paper, we present ARLS (After-action Reviewer for Large-scale Simulations), a MapReduce-based output analysis tool for simulation outputs. ARLS clusters distributed storages using Hadoop and automatically composes Map and Reduce functions to process the simulation outputs. ARLS has been applied to our SAM (Surface-to-Air Missile) simulator. The SAM simulator has been developed to analyze the dynamics of a missile in designing air-defense systems. ARLS takes a large amount of unstructured simulation outputs from SAM simulator, automatically generates Map and Reduce functions to analyze the missile and the aircraft component of SAM simulator, and executes Map and Reduce jobs in parallel. The results of our experiments show that ARLS can efficiently analyze a large amount of unstructured simulation datasets by distributing datasets and computations over the cluster of commodity machines.

1. Introduction

As simulations have come into a wide use in the analysis of large-scale systems with complex structures and dynamics, more engineers have experienced difficulties in interacting with and understanding massive amounts of simulation datasets. For example, simulations on the formation and evolution of large scale structures in the universe used 7.5 million CPU hours and generated 50 terabytes of raw data with additional 25 terabytes of post-processing information [1]. In the domain of catastrophic event risk, millions of simulations need to be quickly performed and large amounts of environmental datasets need to be rapidly analyzed [2]. With the increase of the number of Internet users and its applications, communication network simulators face many challenges related to a rapid analysis of growing teletraffic datasets. The needs of collecting, storing, managing, and analyzing big traffic datasets become imminent for teletraffic simulators to detect and analyze anomaly teletraffic datasets [3]. Defense and military simulations also confront difficulties in analyzing immense datasets coming from live, virtual, and constructive simulators interoperated over the network [4,5]. Therefore, an efficient mechanism is necessary to analyze large-scale simulation outputs and to evaluate the capabilities of the defense system under investigation.

In order to expedite the simulation-based analysis for large-scale systems, it is necessary to speed up not only the amount of time *during* simulations, but also the one *after* simulations. However, less attention has been paid to handling the performance required in the course of analyzing outputs *after* simulations [6]. On the other hand, parallel database management systems, such as Oracle [7], DB2 [8], Teradata [9], and Greenplum [10], as well as new types of large-scale data processing platforms, such as MapReduce and Hadoop [11], impact many scientific and engineering disciplines by providing high-performance computing capabilities in various applications, including bioinformatics, medical science, environmental engineering, e-commerce, manufacturing, and telecommunications.

While the parallel database management systems perform well to process large-scale datasets, they require that the data should conform to a well-defined schema. Unfortunately, many legacy simulators may not produce simulation outputs in a well-structured manner. For example, due to security reasons, legacy missile simulators tend to be available only in a binary format, dumping a large amount of unstructured simulation logs. Since we cannot access the implementation details of the black-box legacy missile simulators, we may need to go through pre-processing steps in order to transform the unstructured simulation logs into

* Corresponding author. Tel.: +82 31330 6444.
E-mail address: ksl@mju.ac.kr (K. Lee).

http://dx.doi.org/10.1016/j.advengsoft.2016.01.025
0965-9978/

well-structured datasets before storing them in parallel database systems. On the other hand, NoSQL solutions, such as Hadoop and MapReduce, permit data to be in any format or even to have no structure at all. Therefore, we can directly handle large-scale and unstructured simulation datasets with Hadoop and MapReduce, without going through additional pre-processing steps. Hadoop is also sufficiently flexible to free us from defining storage requirements in advance. Rather than having a fixed number of data storages, as in the case of parallel database systems, Hadoop and MapReduce platforms allow for scaling the data storages with the increase of the size of outputs.

In this paper, we first illustrate how we can utilize Hadoop and MapReduce to analyze large-scale simulation outputs, with an example of the SAM simulator. The SAM simulator has been developed by us to analyze the dynamics of a missile in designing air-defense systems [12]. The preliminary results obtained from our experiments of SAM simulator suggest that Hadoop and MapReduce model can successfully process the unstructured simulation logs and expedite the analysis process by scaling the required computations over multiple nodes of a cluster. Motivated by these experimental results, we have implemented ARLS, a Hadoop and MapReduce-based output analysis tool, for large-scale simulation logs in any formats. ARLS comprised 8 networked storages clustered by Hadoop and analyzes the large-scale simulation outputs by using the MapReduce computation model. ARLS takes unstructured simulation outputs and stores them in a collection of partitions in HDFS (Hadoop Distributed File System). According to the output measurements a user wants to analyze, ARLS automatically generates Map and Reduce functions. These functions are then injected into the distributed processing framework and are executed to produce the desired analysis results. Results are delivered to users in texts, graphs, and data files.

This paper is organized as follows. Section 2 presents related research and discusses relevant issues in the analysis of large-scale simulation outputs. Section 3 outlines our preliminary research results to show performance improvements by conducting the Hadoop and MapReduce-based simulation output analysis. Section 4 presents ARLS and illustrates how we can utilize ARLS in analyzing large-scale simulation outputs of the SAM simulator. We conclude in Section 5 with a summary and a discussion of further research.

2. Related research

Parallel computing is a type of computation in which multiple computing resources, such as cores in CPU or network-connected computers, simultaneously and concurrently run operations to solve a computational problem [13]. Although parallelism has been used for many years for high-performance computing, the importance of parallel computing has been increasing as large computer clusters become increasingly available. However, due to their high complexity, parallel computing programs are usually more difficult to write than sequential ones. MapReduce was developed by Google [11] as a programming model and an associated implementation to facilitate processing and generating large datasets with a parallel, distributed algorithm on a cluster of commodity hardware. Apache Hadoop was created as an open-source project to make a way to use MapReduce outside Google and now it is becoming a primary platform for processing large amounts of data. Over a half of the Fortune 50 companies are using Hadoop [14]. MapReduce is useful in various applications, including web log analysis [15] and machine learning [16]. One of the most well-known use case of MapReduce is Google's web-search index-building system [17]. The finance industry has also actively adopted Hadoop and MapReduce. JPMorgan Chase, for example, uses Hadoop technology for fraud detection [18]. Morgan Stanley made a scalable portfolio analysis system based on Hadoop and MapReduce [19]. Moreover, MapReduce has already played an important role in many tasks in bioinformatics and life sciences [20], such as genome analysis [21], DNA sequencing [22], and drug development [23].

Unlike the massive enthusiasm around the MapReduce paradigm for large-scale data analysis in various applications, limited research has addressed applying the MapReduce paradigm to simulations and analysis on large-scale engineering systems. In [24], the authors pointed out that traditional molecular dynamics simulation running on parallel programming platforms, such as MPI, OpenMP, or Condor, have a limited capability to schedule the failed sub-tasks to another computing node to implement fault-tolerance. Instead, they used the distributed programming model MapReduce in a small Hadoop cluster and showed that this approach could achieve scalability and fault-tolerance to simulate the molecular dynamics of liquid argon. They also demonstrated that the simulation time could be speed up 28 times when the number of argon particles increased to 108,000. In [1], the authors applied the Hadoop system to cosmological simulations. In analyzing massive astrophysical datasets, ranging from 55 GB to a few TB, it was found that the Hadoop system offers a consistent speedup to process selection and correlation queries. Also, this study confirmed that the Hadoop systems could perform well as the cluster becomes sufficiently large. In [25], the authors proposed a cloudification method based on the MapReduce paradigm to migrate scientific simulations into the cloud to provide a greater scalability. They analyzed its viability by applying it to a real-world railway power consumption simulator and showed that the MapReduce paradigm could be suitable for resource intensive simulations and multidimensional analysis.

Based on the previous research overviewed above, our claim is that MapReduce framework provided by Hadoop is well suited to address the following research issues relevant for the analysis of large-scale simulation outputs:

- To deal with unstructured outputs: Large-scale simulations frequently involve legacy simulators. Since legacy simulators usually do not allow analysts to access implementation details, analysts usually go through pre-processing steps in order to translate the unstructured outputs into a common format for analysis purposes. Unfortunately, the translation process is expensive and, sometimes, cannot be undertaken, if we cannot define a schema on the outputs from legacy simulators. Hadoop is flexible in terms of accepting data of any type, regardless of structure; therefore, it can be efficiently used in interoperating with legacy simulators.
- To deal with scalability of simulation outputs: Simulation-based analysis is often conducted by many experts coming from different disciplines. The kinds and resolutions of outputs could be dynamically changed, as interesting events randomly occur during simulations. Therefore, it is difficult to predict the amount of simulation outputs in advance. Rather than having a fixed number of high-end data storages and servers, it is better to line up a large number of low-end storages and servers and gradually scale them up as the size of outputs becomes large, as in the case of Hadoop.
- To support complex output analysis: Output measurements have a wide spectrum of complexity, ranging from simple statistics to complex values. Map and Reduce functions have the full generality and offer flexibility to express any output analysis algorithms. This feature is particularly beneficial in terms of computing complex and customized output measurements that typical database queries and their combinations are not sufficient to answer.

This paper presents the design and implementation of ARLS, a Hadoop and MapReduce-based simulation output analyser.