MapReduce based Scientific Data Experiment Framework for Transforming Data Set

Yunhee Kang[1], and Heeyoul Choi[2]

[1] Division of Information and Communication
Baekseok University
115 Anseo Dong, Cheonan, Korea 330-704 yunh.kang@gmail.com

[2] Samsung Advanced Institute of Technology
Samsung Electronics.
San 14, Nongseo-dong, Yongin, Korea 446-712
heeyoul@gmail.com

Abstract. Since data volumes in scientific applications have grown exponentially, new scientific methods to analyze and organize the data are required. Hence these methods need to support effective infrastructure composed of computing resources that are used for pre-processing and post-processing data. The demanding requirement has led to development of new programming models and their implementations like MapReduce. In this paper, we describe a framework to support data transformation, which is an essential phase to handling large scale data in scientific data experiments. The framework introduces a way to process a raw data set about tornado outbreak in the US by Hadoop, a MapReduce framework in a parallel manner and it can be applied to pre-processing and post-processing in scientific data experiments.

Keywords: MapReduce, Scientific Data Experiment, data transformation

1 Introduction

Data volumes are approximately doubling each year [1]. Many scientific applications require processes for handling data that no longer fit on a single cost-effective computer. Besides scientific data experiments such as simulations are creating vast data stores that require new scientific methods to analyze and organize the data. Those data experiments for handling scientific problems require the analysis of large amounts of data. Hence these challenges in solving problems need to support effective infrastructure composed of computing resources that are used for pre-processing and post-processing data as well as analyzing data [8]. Those scientific applications devote most of their processing time to I/O and movement of data. Parallel/distributed processing of data-intensive applications typically involves partitioning or subdividing the data into multiple segments which can be processed independently using the same executable application program in parallel on an appropriate computing platform, then reassembling the results to produce the completed output

data [2,8]. A MapReduce programming is able to focus on the problem that needs to be solved since only the map and reduce functions need to be implemented, and its framework takes care of the burden a programmer has to deal with lower-level mechanisms to control the data flow [3,4,5].

In this paper, we describe a design of a framework to support data transformation, which is an essential phase to handling data on scientific experiments. We report how to design a MapReduce application to transform raw data into another one. This approach provides a new way to process raw data by Hadoop, a MapReduce framework in a parallel manner [7].

The rest of paper is organized as follows: Section 2 describes the related works including a brief overview of MapReduce and Hadoop. Section 3 describes an overall architecture of the scientific data framework proposed. The designed MapReduce application is described in section 4. Conclusions are presented in Section 5.

2 Related Works

MapReduce is an emerging programming model for a data-intensive application proposed by Google. MapReduce borrows ideas from functional programming, where programmer defines Map and Reduce tasks to process large set of distributed data. The key strengths of MapReduce programming model are the high degree of parallelism combined with the simplicity of the programming model and its applicability to a large variety of application domains [3,5]. This requires dividing the workload across a large number of machines. The degree of parallelism depends on the input data size. Map function processes the input pairs (key1, value1) returning some other intermediary pair (key2, value2). Then the intermediary pairs are grouped together according to their key. After, each group will be processed by the reduce function which will output some new pairs of the form (key3, value3).

Hadoop [6,7] is an open source based on MapReduce framework for running applications on large clusters built of commodity hardware from Apache. The Hadoop framework transparently provides applications both reliability and data motion. Hadoop implements Map/Reduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. In addition, it provides the Hadoop distributed file system (HDFS) that stores data on the compute nodes, providing a very high aggregate bandwidth across the cluster. HDFS is the primary storage system used by Hadoop applications.

3 Scientific data experiment framework

We describe an overall framework to help building a MapReduce application, which supports to transform data in the field of scientific data experiments and its main considerations. The framework makes sure to seamlessly integrate pre-processing and post-processing phases with data analysis application. The designed framework for scientific data experiment consists of three entities: scientific experimental workspace, science data farm and domain data repository. In the scientific

experimental workspace, a researcher who has a plan of a scientific data experiment defines a workflow of this experiment that consists of processes with a set of activities. Each activity is a single process block that can be linked to another process block if control or data dependence exists between them. An activity might be something as straightforward as a data transformation that extracts specific values in a raw dataset. It can ease a researcher to using large-scale back-end computational resources such as HPC and cloud computing service.

In a scientific data experiment, a workflow can be helpful to organize tasks of the data experiment. When a researcher does a scientific data experiment like simulation, other researcher working in a different domain may clearly define variables that can be ambiguous. In defining phase of a scientific data experiment, ontology is referred to resolve conflict of variables' unit and usage as well as ambiguity of them. According to the workflow defined in the scientific experimental workspace, a scientific data experiment is progressed on the scientific research data farm. The intermediate and final data are moved to the domain data repository. In the domain data repository researchers are sharing data, expertise and knowledge by user portal. The portal is designed to aggregate multiple information sources and applications to provide uniform, seamless, and personalized access for its users. Fig. 1 shows the overall framework for the scientific data experiments.

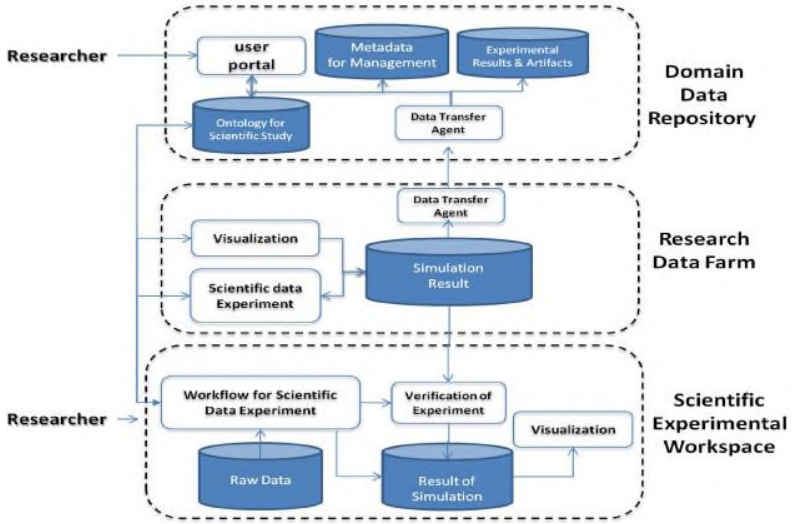

Fig.1. The overall framework for scientific data experiments

In the post-processing phase, data generated during a simulation run are typically validated, subjected to some initial processing and formatting and annotated with appropriated auxiliary information (i.e, metadata) to form a data collection. The type of post-processing varies with the application such as applying scientific codes to interpret the data in fusion applications or generating multi-resolution representations of the data for visualization. In some cases the amount of data generated in the post-

processing phase is equal to or greater than the original data, especially if indexes are built for use in the analysis phase. Post-processing of the data can be done at multiple computational sites. Thus large subsets of the original data can be moved to those computational sites and can be replicated at multiple locations. Different combinations of data from one or more data collections must be accessed and processed to provide the desired result. A typical workflow for scientific data experiment in the application structure is shown in Fig. 2.

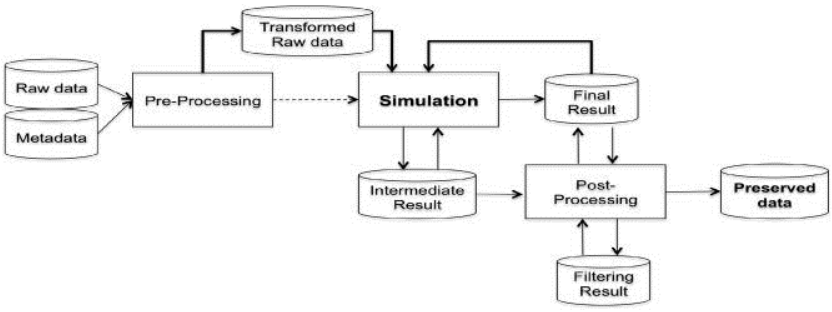

Fig. 2. Basic workflow for scientific data experiment

4 MapReduce Application

In this section, we describe a scientific application to evaluate the proposed framework in which data transformation is applied to extracting a set of data with specific variables. The data transformation in the framework is associated with a pre-processing phase.

4.1 Overview of Application

To apply data transformation of raw data from a dataset to the pre-processing of a scientific data experiment, we construct a MapReduce application that is used to handle a data set collected from the National Climatic Data Center (NCDC, http://www.ncdc.noaa.gov/). The data is stored using a line-oriented ASCII format, in which each line is related to a separate record. The format supports a rich set of meteorological elements. Each record based on the state of the US maintains information about tornado outbreak from 1950 to 1990. It is organized by date with some columns including sequence number, state number, date, weather event, event-remarks, and the number of states involved. We conducted an experimental study to apply data transformation as a part of scientific data experiment to the designed framework. Hadoop v0.21, a MapReduce platform, is used to the experimental study.

4.2 Experimental Results

In the application we counts the number of occurrences of tornado outbreak with regard to year and the state in the US in a dataset. Input key-value pairs take the form of (offset, line) pairs stored on the distributed file system, where the former is a unique identifier for the record, and the later is the record of the data itself. Each line represents a record of data associated with tornado outbreak. The input to the map function is a partition of the data set represented by a multi-dimensional data to process. The map function extracts relevant data, which are associated with the specific variables in a record, by a filter function.

The MapReduce execution framework guarantees that all values associated with the same key are brought together in the reducer. In Hadoop, the reducer is presented with a key and an iterator over all values associated with the particular key. The values are arbitrarily ordered. In Reduce, it needs to sum up all counts associated with a corresponding year. The reduce function calculates the global sum, and then emits the final value with regard to a year.

The input records are mapped as key/value pairs with key being the record id and value being the series record for each year. An intermediate reduce stage is used to construct the series object from the input key/value pairs. The filter function in Map() is designed to extract an intensity value of a tornado from an input line. The Reduce() groups the intermediate values with the key. Fig. 3 shows the result of MapReduce associated with the partial sum of each of intensity of tornado outbreaks.

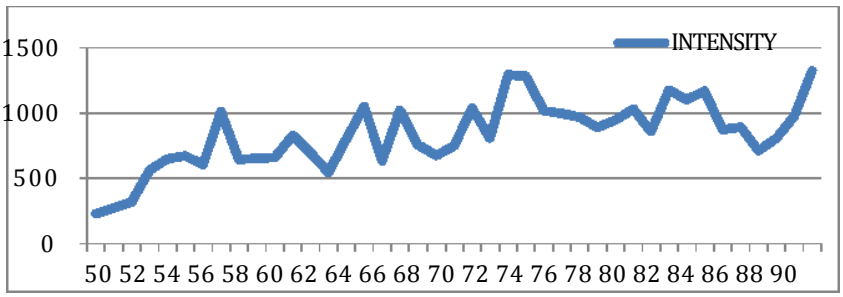

Fig. 3. The result of the MapReduce application represents the intensity of tornado outbreak

5 Conclusion and Future works

New scientific methods to analyze and organize data are required to handle large-scale datasets. Hence these methods need to support effective infrastructure composed of computing resources that are used for pre-processing and post-processing data. We described the design of a framework to support data transformation, which is an essential phase to handling large scale data in scientific data experiments. This

approach introduces a new way to process raw data by Hadoop, a MapReduce framework in a parallel manner. The designed MapReduce application is used to sum up the variables such as tornado occurrence and its intensity.

References

1. J. Gray, et al., "Scientific data management in the coming decade" SIGMOD Rec., vol. 34, pp. 34-41, 2005 2005.2.
2. Han, Y., Bioworks: A Workflow for Automation of Bioinformatics Analysis Processes, International Journal of Bio-Science and Bio-Technology. Vol. 3, No. 4, 59-68(2011)
3. Dean, J., Ghemawat, S.: MapReduce: A Flexible Data Processing Tool. CACM 53, 72-77 (2010)
4. Morton, K., Friesen, A., Balazinska, M., Grossman, D.: Estimating the Progress of MapReduce Pipelines. IEEE 26th International Conference on Data Engineering (ICDE), 2010, pp. 681 - 684, Long Beach, CA (2010)
5. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. CACM 51, 107-113 (2008)
6. F. Wang, et al., "Hadoop high availability through metadata replication," presented at the Proceeding of the first international workshop on Cloud data management, Hong Kong, China, 2009.
7. Apache, http://hadoop.apache.org/
8. Ian Foster, Carl Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2004.