Flexible MapReduce Workflows for Cloud Data Analytics

Carlos Goncalves, ISEL – Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
Luis Assuncao, ISEL – Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
Jose C. Cunha, CITI – FCT, Universidade Nova de Lisboa, Lisbon, Portugal

ABSTRACT

Data analytics applications handle large data sets subject to multiple processing phases, some of which can execute in parallel on clusters, grids or clouds. Such applications can benefit from using MapReduce model, only requiring the end-user to define the application algorithms for input data processing and the map and reduce functions, but this poses a need to install/configure specific frameworks such as Apache Hadoop or Elastic MapReduce in Amazon Cloud. In order to provide more flexibility in defining and adjusting the application configurations, as well as in the specification of the composition of the application phases and their orchestration, the authors describe an approach for supporting MapReduce stages as sub-workflows in the AWARD framework (Autonomic Workflow Activities Reconfigurable and Dynamic). The authors discuss how a text mining application is represented as a complex workflow with multiple phases, where individual workflow nodes support MapReduce computations. Access to intermediate data produced during the MapReduce computations is supported by a data sharing abstraction. The authors describe two implementations of this abstraction, one based on a shared tuple space and another based on an in-memory distributed key/value store. The authors describe the implementation of the framework, a set of developed tools, and their experimentation with the execution of the text mining algorithm over multiple Amazon EC2 (Elastic Compute Cloud) instances, and report on the speed-up and size-up results obtained up to 20 EC2 instances and for different corpus sizes, up to 97 million words.

Keywords: Cloud, Data Analytics Applications, MapReduce, Text Mining, Workflow

INTRODUCTION

In order to support the requirements for processing, storage and data transfer requirements of data-intensive applications, there has been recent research on parallel programming models, support tools, and runtime environments for cluster, grid and cloud computing platforms.

In the field of information retrieval and data analytics there is a need to automatically select and extract relevant information from large-scale document collections. In this paper we report on the development of a parallel execution framework for a text mining algorithm that performs the automatic extraction of relevant expressions from text corpora. These applications can be expressed as workflow structures with multiple nodes for data pre-processing, analysis, combination and visualization, thus requiring the flexible composition of the work-
flow nodes. The parallelization of individual workflow nodes and their composition are both illustrated in the case of the text mining application presented in this paper. Both aspects were implemented in a unified workflow framework, AWARD (Autonomic Workflow Activities, Reconfigurable and Dynamic) previously proposed in (Assuncao et al., 2012), which was used for expressing the logical application workflow and for launching its execution on the Amazon Elastic Compute Cloud (EC2) infrastructure.

In this paper we discuss the design and implementation of the parallelization of individual nodes of the text mining workflow using MapReduce, a programming model for data-parallel processing of large data sets. The implicit MapReduce workflow template is automatically instantiated by the runtime system with pre-defined stages for input data splitting, parallel evaluation of Map, partitioning and distributing the intermediate data, and parallel evaluation of Reduce. By supporting MapReduce workflows on top of the AWARD framework, we achieved increased flexibility concerning the workflow composition and its configuration according to the application requirements (Goncalves et al., 2012).

Furthermore, we discuss two implementations of an abstraction for data sharing that was used to store the intermediate key/value pairs produced by Mappers, and retrieved by the Reducers: i) An implementation based on the concept of the Linda tuple space (Carriero & Gelernter, 1989) where data (tuples) are stored and retrieved by pattern matching; ii) Another implementation based on in-memory distributed key/value store supporting multiple values for each key. In order to deploy large and complex workflows on multiple EC2 instances, we also developed a set of tools providing support for: i) Specifying and configuring AWARD workflows; ii) Instantiating Amazon EC2 instances; iii) Starting the workflow execution; iv) Monitoring the workflow execution.

We present the experimental results from the execution of the text mining algorithm on Amazon EC2. As results we report on the speed-up and size-up achieved for different execution scenarios of virtual machine (EC2) instances and for different corpus sizes from 16 to 97 millions of words.

Firstly, we review related work, and then we describe the text mining application and its workflow. The AWARD framework and the support for MapReduce workflows are explained next, including the description of the two implementations of the data sharing abstraction. Then we describe the developed tools for workflow configuration and mapping. Finally we present the experimental results obtained on the Amazon EC2 infrastructure and we discuss conclusions and future work.

RELATED WORK

Recent research in data analytics applications (Melnik et al., 2011) is influencing the developments on programming models and support tools and environments. The need to process large-scale data sets requires the runtime capability for exploring data-parallelism automatically and dynamically. There is also a requirement for application decomposition in functionally specialized processing stages, with efficient data transfer and communication between stages.

These concerns have been addressed through several approaches based on the MapReduce model (Dean & Ghemawat, 2004), which emerged as a widely used programming model, due to its conceptually simple programming interface for specifying the input data format, and the Map and Reduce functions, and for enabling efficient implementations on different platforms and architectures (Dean & Ghemawat, 2004; Apache Hadoop, 2012; Grossman & Gu, 2008; Amazon EMR, 2012; Riteau et al., 2011; Gunarathne et al., 2010; He et al., 2008).

Several attempts have been made to extend its expressiveness (Ekanayake et al., 2010), by supporting iterations and incremental computations (Bhatotia et al., 2011, Peng and Dabek, 2010), higher level data abstractions (Gates et al., 2009, Isard et al., 2007, Thusoo et al., 2009), and database concepts (Abouzeid et al.,...
Speech Enhancement Using Heterogeneous Information
www.igi-global.com/article/speech-enhancement-using-heterogeneous-information/205503?camid=4v1a