Multisite Management of Data-intensive Scientific Workflows in the Cloud

Ji Liu

Multisite Management of Data-intensive Scientific Workflows in the Cloud

Ji Liu
ji.liu@inria.fr
Thesis Start Date: 01/10/2013
MSR-INRIA Joint Centre, Paris, France
LIRMM, University Monpellier 2, Montpellier, France

1. INTRODUCTION

Scientific experiments generally contain multiple computational activities to process experimental data and these activities are related by data or control dependencies. Scientific workflows enable scientists to model the data processing of these experiments as a graph, in which vertexes represent data processing activities and edges represent dependencies between them. Since scientific workflow activities may process big data, one scientific workflow activity can correspond to several executable tasks for different parts of input data during scientific workflow execution. As big amounts of data are handled or produced during the scientific workflow execution, data-intensive scientific workflows become an important issue.

A Scientific Workflow Management System (SWfMS) is an efficient tool to execute scientific workflows. In order to execute a data-intensive scientific workflow within reasonable time, SWfMSs generally exploit High Performance Computing (HPC) resources in a cluster, grid or cloud environment. Because of virtually infinite resources, diverse scalable services, stable service quality and flexible payment policies, cloud become a primary solution for workflow execution.

Due to the geographic distribution of scientists, data and computing resources, multisite cloud is appealing for data-intensive scientific workflow execution. A multisite cloud contains multiple data centers in multiple cloud sites and each data center is explicitly accessible to cloud users. Explicitly accessible has two meanings. The first meaning is that each cloud site is separately visible and accessible to cloud users. The other meaning is that cloud providers will not change the data location, i.e. the data of users will not be moved from one cloud site to another cloud site without the permission of users. In this case, SWfMSs need to meet the challenges of scheduling data and executable tasks to computing nodes across multiple cloud sites. In addition, SWfMSs have to efficiently transfer data within one site or across different sites at this situation. The main objective of this thesis is to propose efficient approaches to execute data-intensive scientific workflows in a multisite cloud.

BDA 14 octobre 2014, Grenoble-Autrans, France.

The current solutions for the parallel execution of scientific workflows are appropriate for static computing and storage resources in a grid environment. They have been extended to deal with more elastic resources in a cloud, but with only one site. Our analysis [1] of the current techniques of scientific workflow parallelization and scientific workflow execution has shown that there is a lot of room for improvement in the following directions:

1. Data staging: existing techniques mainly focus on the mechanism that starts scientific workflow execution after gathering all the related data in a shared-disk file system at one data center, which is time consuming.

2. Architecture: the structure of SWfMSs is generally centralized, with a master node, which is a single point of failure and performance bottleneck, managing all the optimization and scheduling processes.

3. Task scheduling and data location: most SWfMSs do not take data location into account during task scheduling, which makes it inefficient to read or write data.

4. Multisite: novel task and data scheduling approaches are required for utilizing resources in a multisite cloud.

In the rest of this paper, we define more precisely the problem and introduce our approach to address it.

2. PROBLEM DEFINITION

We consider the problem of executing scientific workflows in a multisite cloud. A scientific workflow representation file is stored in a Virtual Machine (VM) of a cloud site while workflow data may be distributed in different cloud sites because of the geographical distribution of scientists. The workflow data includes the data to be processed by a scientific workflow and the instruction data, i.e. the instructions of the programs in the scientific workflow. Some data can be transferred between different sites while some other data is fixed at a specific site and cannot be moved to another site because of big data or security restrictions.

Generally, the execution of scientific workflows is realized by the VMs at each site. The VMs in each site are able to communicate with each other with appropriate configuration, e.g. public IP address, port number etc. The data transfer rate, however, is different in different situations, e.g. intersite communication, intrasite communication, the VMs that share storage resources and so on. Moreover, the VMs and required storage resources can be created before

workflow execution or dynamically created during workflow execution.

We formulate the problems we address in this thesis as follows. Given a scientific workflow W, which is composed of activities V and dependencies E, and a multisite cloud MS, in which each site stores its own data (input data for W or instruction data of W) and there are available computing and storage resources to execute the scientific workflow, how to optimize and schedule W in MS in a way that achieves required objectives while respecting additional limitations, e.g. security restrictions, budget limitations etc. This problem can be divided into two sub-problems:

Workflow Parallelization Given a scientific workflow $W = \{V,E\}$ and a multisite cloud $MS = \{S_1, S_2, ..., S_n\}$, how to efficiently parallelize scientific workflow execution and schedule each part of W into available VMs at each site with consideration of activity dependencies in E and special features of each site to achieve required objectives, e.g. minimizing execution time, reducing monetary cost etc.

Data Management Given the data in a scientific workflow, how to manage it in a multisite management environment to achieve required objectives, e.g. minimizing the time to get data or reducing data storage cost etc.

The data can be stored in the local disk of a VM, a global file system for all the VMs at one site or a global file system for all the VMs at multiple sites. Thus, it is important to generate appropriate strategies to transfer and store data among different cloud sites during scientific workflow execution. In addition, since there are big data produced during workflow execution, it is also helpful to support dynamic elastic storage resource provisioning, i.e. data storage resources can be inserted to or removed from VMs during workflow execution.

3. PROPOSED APPROACH

Workflow parallelization is the process of transforming and optimizing a (sequential) workflow into a parallel WEP, which consists of workflow parallelism and workflow scheduling. The WEP is a program that captures optimization decisions and execution directives, typically the result of compiling and optimizing a workflow. Similar to the concept of Query Execution Plan (QEP) in distributed database systems [6], it allows the SWfMS to execute scientific workflows in parallel in multiple computing nodes.

A SWfMS can exploit activity parallelism or data parallelism to execute a scientific workflow. Activity parallelism makes the execution of different activities run on different computing nodes at the same time. Activity parallelism includes independent parallelism and pipeline parallelism. Independent parallelism achieves parallel execution for the activities that have few data or control dependencies between them. Pipeline parallelism is for executing dependent activities, where one may process the output data of another, in different computing nodes at the same time. Data parallelism is obtained by having multiple tasks performing the same activity, each on a different data chunks while the tasks can be simultaneously executed in different computing nodes.

A SWfMS can exploit all the possible types of parallelism, i.e. independent parallelism, pipeline parallelism and data parallelism to achieve efficient parallel execution of scientific workflows. First, a SWfMS can partition a scientific workflow into workflow fragments and distribute each workflow fragment into appropriate sites, at which input data is stored. By minimizing data transfer volumes between different workflow fragment, workflow partitioning can yield good performance [2]. Scientific workflow partitioning partially realizes the independent parallelism across multiple cloud sites. Second, the identification of independent activities achieves complete independent parallelism within one cloud site. Third, a SWfMS can achieve pipeline parallelism by allocating dependent activities in different computing nodes and by making them process different parts of data at the same time. Finally, the parallel execution of one activity can be achieved by distributing executable tasks of this activity into different VMs. To achieve these different kinds of parallelism, an algebraic approach is a good solution because of its powerful operators and possible optimization for the entire algebraic workflow expression [4]. Similar to multisite query processing in distributed and parallel database systems [6, 7], we intend to adapt the algebraic approach to a multisite cloud.

After devising a parallel execution strategy for a scientific workflow, a SWfMS should schedule executable tasks to available computing nodes through a static, dynamic or hybrid method, i.e. the combination of static and dynamic methods. We argue that SWfMSs can make a static WEP to specify a static scheduling strategy before execution. During workflow execution, if the execution environment varies a lot and the execution cannot achieve load balancing through the original WEP, the SWfMS can exploit dynamic scheduling according to the run-time environment features. Furthermore, SWfMSs can dynamically insert or remove VMs to achieve multiple objectives, e.g. budget limit, time limit etc [5].

For managing data, most existing SWfMSs exploit a shared-disk file system and perform small optimization to store data. In order to achieve better performance, we propose to schedule tasks according to data location while using local storage resources across the VMs in multiple cloud sites for data-intensive scientific workflow execution. Since the activities in a data-intensive scientific workflow may process massive data sets, we believe this co-scheduling method can accelerate scientific workflow execution and avoid useless data transfer cost among different VMs located at different sites.

Furthermore, the data transfer between different sites can be directly achieved through multiple computing nodes at each site while data streams do not need to pass through a centralized computing node during execution. This solution can avoid the communication bottleneck caused by a centralized architecture.

Finally, we plan to validate our solutions with BlobSeer [3], a distributed file system, on Microsoft Azure. We will adapt BlobSeer to multisite cloud and optimize the communication methods between different sites and the cooperation between BlobSeer and SWfMSs.

4. ACKNOWLEDGMENT

The thesis is advised by Esther Pacitti (University of Montpellier), Patrick Valduriez (INRIA) at LIRMM and Marta Mattoso (UFRJ).

The work is partially funded by Microsoft (Zcloudflow project) and CNPq, FAPERJ, INRIA (HOSCAR and MUSIC projects, and performed in the context of the Computational Biology Institute (`www.ibc-montpellier.fr`).

5. REFERENCES

[1] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso. A survey of data-intensive scientific workflow management. *Journal of Grid Computing*, 2015. To appear.

[2] J. Liu, V. Silva, E. Pacitti, P. Valduriez, and M. Mattoso. Scientific workflow partitioning in multi-site clouds. In *Euro-Par 2014: Parallel Processing Workshops*, pages 1–12, 2014.

[3] B. Nicolae, G. Antoniu, L. Bougé, D. Moise, and A. Carpen-Amarie. Blobseer: Next-generation data management for large scale infrastructures. *Journal of Parallel and Distributed Computing*, 71(2):169–184, 2011.

[4] E. Ogasawara, D. Oliveira, P. Valduriez, J. Dias, F. Porto, and M. Mattoso. An algebraic approach for data-centric scientific workflows. *Proceedings of the VLDB Endowment (PVLDB)*, 4(12):1328–1339, 2011.

[5] D. Oliveira, K.A.C.S. Ocaña, E. Ogasawara, J. Dias, J. Gonçalves, F. Baião, and M. Mattoso. Performance evaluation of parallel strategies in public clouds: A study with phylogenomic workflows. *Future Generation Computer Systems*, 29(7):1816–1825, 2013.

[6] M. T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Springer, 2011.

[7] E. Pacitti, R. Akbarinia, and M.E. Dick. *P2P Techniques for Decentralized Applications*. Morgan & Claypool Publishers, 2012.