Future Generation Computer Systems 26 (2010) 1200–1214

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A data placement strategy in scientific cloud workflows

Dong Yuan *, Yun Yang, Xiao Liu, Jinjun Chen

Faculty of Information and Communication Technologies, Swinburne University of Technology, Hawthorn, Melbourne 3122, Australia

ARTICLE INFO

Article history:
Received 30 May 2009
Received in revised form
26 November 2009
Accepted 1 February 2010
Available online 18 February 2010

Keywords:
Data management
Scientific workflow
Cloud computing

ABSTRACT

In scientific cloud workflows, large amounts of application data need to be stored in distributed data centres. To effectively store these data, a data manager must intelligently select data centres in which these data will reside. This is, however, not the case for data which must have a fixed location. When one task needs several datasets located in different data centres, the movement of large volumes of data becomes a challenge. In this paper, we propose a matrix based *k*-means clustering strategy for data placement in scientific cloud workflows. The strategy contains two algorithms that group the existing datasets in *k* data centres during the workflow build-time stage, and dynamically clusters newly generated datasets to the most appropriate data centres – based on dependencies – during the runtime stage. Simulations show that our algorithm can effectively reduce data movement during the workflow's execution.

1. Introduction

Running scientific workflow applications usually need not only high performance computing resources but also massive storage [1]. In many scientific research fields, like astronomy [2], high-energy physics [3] and bio-informatics [4], scientists need to analyse terabytes of data either from existing data resources or collected from physical devices. During these processes, similar amounts of new data might also be generated as intermediate or final products [1]. Workflow technologies are facilitated to automate these scientific applications. Scientific workflows are typically very complex. They usually have a large number of tasks and need a long time for execution. Nowadays, popular scientific workflows are deployed in grid systems [3] because they have a high performance and massive storage. However, building a grid system is extremely expensive and it is not available for scientists all over the world to use.

The emergence of cloud computing technologies offers a new way to develop scientific workflow systems. Since late 2007 the concept of cloud computing was proposed [5] and it has been utilised in many areas with some success [6–9]. Cloud computing is deemed as the next generation of IT platforms that can deliver computing as a kind of utility [10]. Foster et al. made a comprehensive comparison of grid computing and cloud computing [11]. Some features of cloud computing also meet the requirements of scientific workflow systems. First, cloud computing systems can provide a high performance and the massive storage required for scientific applications in the same way as grid systems, but with a lower infrastructure construction cost among many other features, because cloud computing systems are composed of data centres which can be clusters of commodity hardware. Second, cloud computing systems offer a new paradigm so that scientists from all over the world can collaborate and conduct their research together. Cloud computing systems are based on the Internet, and so are the scientific workflow systems deployed on the cloud. Dispersed computing facilities (like clusters) at different institutions can be viewed as data centres in the cloud computing platform. Scientists can upload their data and launch their applications on scientific cloud workflow systems from anywhere in the world via the Internet. As all the data are managed on the cloud, it is easy to share data among scientists. Research into doing science on the cloud has already commenced such as early experiences like the Nimbus [12] and Cumulus [13] projects. The work by Deelman et al. [14] shows that cloud computing offers a cost-effective solution for data-intensive applications, such as scientific workflows [15].

By taking advantage of cloud computing, scientific workflow systems could gain a wider utilisation; however they will also face some new challenges, where data management is one of them. Scientific applications are data intensive and usually need collaborations of scientists from different institutions [16], hence the application data in scientific workflows are usually distributed and very large. When one task needs to process data from different data centres, moving the data becomes a challenge [1]. Some application data are too large to be moved efficiently, some may have fixed locations that are not feasible to be moved and some may have to be located at fixed data centres for processing, but this is only one aspect of this challenge. For the application data that are flexible to be moved, we also cannot move them whenever and wherever we want, since in the cloud computing platform, data centres may belong to different cloud service providers so that data movement would result in costs. Furthermore, the infrastructure of the

* Corresponding author.
E-mail addresses: dyuan@swin.edu.au (D. Yuan), yyang@swin.edu.au (Y. Yang), xliu@swin.edu.au (X. Liu), jchen@swin.edu.au (J. Chen).

0167-739X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2010.02.004

Denotations

d_i	dataset
D	set of datasets
D_i	set of datasets in a partition
fd_i	fixed location dataset
FD	set of fixed location datasets
t_i	workflow task
T	set of workflow tasks
T_i	set of workflow tasks that will use dataset d_i
dc_i	data centre
DC	set of data centres
p_i	partition of datasets
P	set of partitions
s_i	size of a dataset
cs	size of a data centre
ds	size of a partition
ps	size of a set of partitions
FP	set of partitions that have fixed location datasets
NFP	set of partitions that do not have fixed location datasets
DM	dependency matrix
CM	clustered dependency matrix
CM_i	sub clustered dependency matrix
CM_T	the top sub clustered dependency matrix after one binary partition
CM_B	the bottom sub clustered dependency matrix after one binary partition
GM	global measure of BEA transformation
PM	global measure of binary partition
dep_{ij}	dependency between datasets d_i and d_j
dc_dep_{ij}	dependency between dataset d_i and data centre dc_j
K	set of data centres with placement of datasets
λ_{ini}	initial storage usage parameter of data centres
$\lambda_{\max}$	maximum storage usage parameter of data centres

cloud computing systems is hidden from their users. They just offer the computation and storage resources required by users for their applications. The users do not know the exact physical locations where their data are stored. This kind of model is very convenient for users, but remains a big challenge for data management to scientific cloud workflow systems.

In this paper, we propose a matrix based k-means clustering strategy for data placement in scientific cloud workflow systems. Scientific workflows can be very complex, one task might require many datasets for execution; furthermore, one dataset might also be required by many tasks. If some datasets are always used together by many tasks, we say that these datasets are dependant on each other. In our strategy, we try to keep these datasets in one data centre, so that when tasks were scheduled to this data centre, most, if not all, of the data they need are stored locally.

Our data placement strategy has two algorithms, one for the build-time stage and one for the runtime stage of scientific workflows. In the build-time stage algorithm, we construct a dependency matrix for all the application data, which represents the dependencies between all the datasets including the datasets that may have fixed locations. Then we use the BEA algorithm [17] to cluster the matrix and partition it that datasets in every partition are highly dependent upon each other. We distribute the partitions into k data centres, where the partitions have fixed location datasets are also placed in the appropriate data centres. These k data centres are initially as the partitions of the k-means algorithm at the runtime stage. At the runtime, our clustering algorithm deals with the newly generated data that will be needed by other tasks. For every newly generated dataset, we calculate its dependencies with all k data centres, and move the data to the data centre that has the highest dependency with it.

By placing data with their dependencies, our strategy attempts to minimise the total data movement during the execution of workflows. Furthermore, with the pre-allocation of data to other data centres, our strategy can prevent data gathering to one data centre and reduces the time spent waiting for data by ensuring that the relevant data are stored locally.

The remainder of the paper is organised as follows. Section 2 presents the related work. Section 3 gives an example and analyses the research problems. Section 4 introduces the basic strategy of our algorithms. Section 5 presents the detailed steps of the algorithms in our data placement strategy. Section 6 demonstrates the simulation results and the evaluation. Finally, Section 7 addresses our conclusions and future work.

2. Related work

Data placement of scientific workflows is a very important and challenging issue. In traditional distributed computing systems, much work about data placement has been conducted. In [18], Xie proposed an energy-aware strategy for data placement in RAID-structured storage systems. Stork [19] is a scheduler in the Grid that guarantees that data placement activities can be queued, scheduled, monitored and managed in a fault tolerant manner. In [20], Cope et al. proposed a data placement strategy for urgent computing environments to guarantee the data's robustness. At the infrastructure level, NUCA [21] is a data placement and replication strategy for distributed caches that can reduce the data's access latency. However, none of them focuses on reducing the data's movement between data centres on the Internet. As cloud computing has become more and more popular, new data management systems have also appeared, such as the Google File System [22] and Hadoop [23]. They all have hidden infrastructures that can store the application data independent of the users' control. The Google File System is designed mainly for Web search applications, which are different from workflow applications. Hadoop is a more general distributed file system, which has been used by many companies, such as Amazon and Facebook. When you push a file to a Hadoop File System, it will automatically split this file into chunks and randomly distribute these chunks in a cluster. Furthermore, the Cumulus project [13] introduced a scientific cloud architecture for a data centre. And the Nimbus [12] toolkit can directly turn a cluster into a cloud and it has already been used to build a cloud for scientific applications. Within a small cluster, data movement is not a big problem, because there are fast connections between nodes, i.e. the Ethernet. However, the scientific cloud workflow system is designed for scientists to collaborate, where large scale and distributed applications need to be executed across several data centres. The data movement between data centres may cost a lot of time, since data centres are spread around the Internet with limited bandwidths. In this work, we try to place the application data based on their dependencies in order to reduce the data movement between data centres.

Data transfer is a big overhead for scientific workflows [24]. Though popular scientific workflow systems have their data management strategies, they do not focus on reducing data movement. For the build-time stage, these systems mainly focus on the data modelling methods. For example, Kepler [3] has an actor-oriented data modelling method that works for large data in a grid environment, Taverna [4] and ASKALON [25] have their own process definition language to represent their data flows. For the runtime stage, most of the scientific workflow systems adopt some data grid systems for their data management. For examples, Kepler uses the SRB [26] system, while Pegasus [2] and Triana [27] adopt the RLS