Future Generation Computer Systems 37 (2014) 321–334

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Pareto frontier for job execution and data transfer time in hybrid clouds

Javid Taheri [a,*], Albert Y. Zomaya [a], Howard Jay Siegel [b], Zahir Tari [c]

[a] School of Information Technologies, The University of Sydney, Australia

[b] Department of Electrical and Computer Engineering, Colorado State University, United States

[c] School of Computer Science, RMIT University, Australia

HIGHLIGHTS

- Particle Swarm Optimization to find the Pareto frontier of data-aware job scheduling.
- Pareto frontier of execution of jobs vs. transfer time of their required data-files.
- Analysis of the influence of Big-data and/or Private-data presence in hybrid clouds.
- Significant outperformance in comparison with current algorithms.
- Fast convergence speed; usually a few minutes for typical hybrid clouds.

ARTICLE INFO

Article history:
Received 10 July 2013
Received in revised form
14 October 2013
Accepted 3 December 2013
Available online 18 December 2013

Keywords:
Big data
Private data
Cloud bursting
Particle swarm optimization
Pareto frontier

ABSTRACT

This paper proposes a solution to calculate the Pareto frontier for the execution of a batch of jobs versus data transfer time for hybrid clouds. Based on the nature of the cloud application, jobs are assumed to require a number of data-files from either public or private clouds. For example, gene probes can be used to identify various infection agents such as bacteria, viruses, etc. The heavy computational task of aligning probes of a patient's DNA (private-data) with normal sequences (public-data) with various data sizes is the key to this process. Such files have different characteristics – depends on their nature – and could be either allowed for replication or not in the cloud. Files could be too big to replicate (big data), others might be small enough to be replicated but they cannot be replicated as they contain sensitive information (private data). To show the relationship between the execution time of a batch of jobs and the transfer time needed for their required data in hybrid cloud, we first model this problem as a bi-objective optimization problem, and then propose a Particle Swarm Optimization (PSO)-based approach, called here PSO-ParFnt, to find the relevant Pareto frontier. The results are promising and provide new insights into this complex problem.

1. Introduction

Cloud computing is a service oriented computing paradigm that has significantly revolutionized computing through its many services – Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) – as well as some of the recently added ones: Database as a Service and Storage as a Service. A large number of application domains have leveraged such services and provided a variety of cloud-based solutions [1]. As a result of such a shift, data have been produced and consumed at much higher rates when compared to traditional grid or cluster systems. Scalable job scheduling and database management systems for both CPU-intensive workloads as well as data-intensive applications have thus become a critical part of current cloud infrastructures [1].

Along with public clouds (e.g., Microsoft Azure [2], and Amazon EC2 [3]), many companies have constructed their own private cloud infrastructure through transforming many of their legacy systems. Although having a private cloud is an advantage for many organizations, sudden needs for extra computing capabilities might lead some of these organizations to outsource portions of their computation needs. This leads to another application development model, known as *cloud bursting*, where an application is run in a private cloud and bursts into (i.e. expands into) a public cloud should the demand for computing exceed available resources. Experts, however, recommend cloud bursting only for non-sensitive applications, for example, those applications that do not require private/sensitive data to run. Because of such security issues, organizations tend to use their private clouds even when performing all computation in public clouds is cheaper. It is also believed that cloud bursting works best when either an applica-

* Corresponding author. Tel.: +61 290369718.
E-mail addresses: javid.taheri@sydney.edu.au (J. Taheri), albert.zomaya@sydney.edu.au (A.Y. Zomaya), HJ@ColoState.edu (H.J. Siegel), zahir.tari@rmit.edu.au (Z. Tari).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.12.020

tion does not have complex interdependency with other applications, or when applications are moved to public clouds so that local resources are spared for more business-critical applications [1].

Big-data is another reason why many computations must be performed externally to one's private cloud. Although the definition of big-data has not been fully agreed upon yet, it is always used to describe a voluminous amount of unstructured or semi-structured data, usually on the order of terabytes and beyond, created from one or multiple sources. Big-data is usually defined using the following "4-V's" [4]: Volume, Variety, Velocity, and Variability. *Volume* refers to data that is large in size; *Variety* refers to a data set composed of many sources and which is probably unstructured; *Velocity* refers to change of the data rate coming to a process; and *Variability* refers to the fact that sometimes it is almost impossible to predict value of information that may come to you tomorrow. All these V's imply that computation must be usually performed where the data resides; Volume of data also further restricts it to a no-replication policy for big-data sometimes. However, from the security point of view, organizations may still decide to replicate big-data on their private clouds for further analysis because they might not be able to tolerate delays of such large transfers from public clouds to their private infrastructure.

From the scheduling point of view, providing solutions that consider all the aforementioned restrictions (i.e. security for private data and size for big-data) and efficiently execute a batch of jobs in a hybrid (private plus public) cloud is far more difficult than the original data-dependent job scheduling problem in grids. In fact, such solutions must consider not only location of data-files in addition to computational capacity of clouds in scheduling decisions, but also the privacy and unusual size of few very large-sized data-files in a system. Because of such extra difficulties in dealing with both complex restrictions, many proposed schedulers of such hybrid systems are usually over-simplified to produce the fastest and mostly the simplest solutions.

After close examination of many already proposed schedulers, we noticed that no proper investigation is ever conducted for hybrid clouds to discover the inter-relationship between the execution time of a batch of jobs and the transfer time required to deliver (cache or replicate) their required data when the size of data is large [5–13,15,16]. Prior investigations for grids showed that these two objectives usually contradict each other where minimizing one usually results in compromising the other [13,15,17]. For example, minimizing the execution time of a batch of jobs requires scheduling jobs to clouds with more computing cores, whereas minimizing the transfer time of data requires scheduling jobs to clouds where the needed data already reside.

We have also realized that most of such techniques are usually tailor-made to either minimize the execution time of jobs or the transfer time of all data-files in a system, with very few exceptions that consider both. We also realized that it is impossible to measure the true performance of such algorithms when migrated to clouds without knowing their optimal (either theoretical or computational) scheduling solutions. PSO-ParFnt is a technique we designed to address this issue, because it is designed to computationally find the Pareto frontier of hybrid clouds and reveal the true performance of different algorithms in various situations. All programs that may require cloud-bursting of some or all of their processes can directly benefit from the outcome of PSO-ParFnt to balance the execution time of their jobs versus the amount of data that must be transfered to/from the cloud. Astronomy applications such as Montage [18], bioinformatics applications such as DNA sequencing [19], and climate modeling applications [20] are among many applications with such nature.

This paper proposes an approach to model such complex relationship and analyze its trade-offs. To this end, we first model the problem as a bi-objective optimization problem and then use our proposed Particle Swarm Optimization (PSO) approach to compute the Pareto frontier of the trade-offs. The Pareto fronts for our case studies are then aligned with several already proposed hybrid scheduling algorithms to (1) validate the quality of our computed Pareto fronts, and (2) validate the quality of a few already proposed solutions by measuring their distance from the calculated Pareto fronts.

This work is organized as follows. Section 2 highlights related work followed by preliminaries of the proposed approach in Section 3. Section 4 details the solution for the computation of the Pareto front. Section 5 overviews the simulation setup and details the results of the simulation studies. Section 6 analyses the results and summarizes the important outcomes. Finally, Section 7 concludes our work and highlights future directions for study.

2. Related work

The work in this paper is closely related to three main aspects of cloud computing: big-data transfer complexities, data privacy, and scheduling data dependent jobs in hybrid clouds. Because comprehensive literature reviews for each of these topics are beyond the scope of this study, we provide sufficient details on each aspect by covering only issues directly related to our proposed solutions.

Big-data: In information technology, big-data [21,22] consists of data sets that usually grow too large and become complex to handle using current database management tools; capture, storage, search, share, analytics, and visualization are among some of the well-known issues [23]. Despite its many challenges, the trend of incorporating big-data is still continuing as it has the potential to provide deeper analysis to detect business trends, prevent diseases, combat crime, and others [24]. Data sets are continually growing in size as they are usually collected from a variety of sources, such as ubiquitous information-sensing mobile devices, aerial sensory technologies (remote sensing), software logs, cameras, microphones, radio-frequency identification readers, and wireless sensor networks [25]. As a result, the world's technological capacity to store information has roughly doubled every 40 months [26]. Along with this trend, database requirements are also vastly different from one organization to another. Greenplum [27] is an example of such databases where the emphasis is to provide very fast data loading to other applications. Fig. 1 conceptually shows how different data-file systems can be categorized according to their structure and scaling capabilities [28]. As capacity needs grow, in scale-up storage systems, disks are added behind an already existing storage controller; in scale-out systems, complete storage elements are added to the system. Loosely structured and scale-out architectures are essential and favored for big-data initiatives.

Private-data: Fujitsu conducted a global survey in October 2010 to study consumer attitudes and concerns about having their personal data in a cloud [29]. The survey revealed that although consumers are excited and intrigued by opportunities that arise from cloud computing, they are also deeply concerned about their data privacy and risks involved in sharing data. There have been several legal studies to properly define "personal data" in the cloud [30]. Serious questions remain as to whether databases containing anonymized, pseudoanonymized, encrypted, and fragmented data in transmission and/or storage should still be considered as "private" or not. As a result of this, many service providers, such as financial institutions, prefer not to take the risk of using cloud bursting in order not to compromise the safety of their data.

Data aware job scheduling: For jobs with file dependencies, especially data-intensive ones, scheduling not only involves computational concerns, but also the data management to access required data-files. Data replication techniques have been around for many