J. Parallel Distrib. Comput. 113 (2018) 1–16

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

Dynamic scheduling strategy with efficient node availability prediction for handling divisible loads in multi-cloud systems

Seungmin Kang [a,*], Bharadwaj Veeravalli [a], Khin Mi Mi Aung [b]

[a] Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117583, Singapore
[b] Data Storage Institute, A*STAR, 2 Fusionopolis Way, #08-01 Innovis, Singapore 138634, Singapore

HIGHLIGHTS

- We address the scheduling problem in multi-cloud systems.
- We cope with the uncertainty of node availability by using prediction technique.
- We propose Dynamic Scheduling Strategy (DSS) for multi-cloud systems.
- We demonstrate the effectiveness of DSS through extensive simulations.

ARTICLE INFO

Article history:
Received 29 August 2016
Received in revised form 29 August 2017
Accepted 15 October 2017

Keywords:
Cloud computing
Scheduling strategy
Multi-cloud system
Divisible load theory
Prediction techniques

ABSTRACT

With large resource capacity, clouds have become a primary infrastructure for users to store big amount of data and perform large-scale computations. With the increasing demands and diversity of applications' requirements, users are facing a fundamental problem of management of resources reserved from clouds. Particularly, the problem of load scheduling is the most important since it directly affects the performance of the system. Designing an efficient scheduling strategy for minimizing the total processing time of loads is challenging since it has to consider many intrinsic characteristics of the system such as the availability and heterogeneity of computing nodes, network topology and capacity. In this paper, we propose a novel architecture of a multi-cloud system that can satisfy complex requirements of users' applications. Based on this architecture, we propose a dynamic scheduling strategy (DSS) that integrates the Divisible Load Theory and node availability prediction techniques to achieve high performance. We conduct intensive simulations to evaluate the performance of the proposed scheduling strategy. The results show that the proposed scheduling strategy outperforms the baseline schemes by reducing the total processing time of loads up to 44.60%. The results also provide useful insights on the applicability of the proposed approach in realistic scenarios.

1. Introduction

The advent of cloud computing has changed the way of satisfying computing resource demands of both individual users and organizations. Leveraging on diverse features of clouds such as IaaS and PaaS, cloud users are nowadays exploiting the immense computational and storage capacities of clouds in a flexible manner at a low cost compared to the *total cost of ownership* (TCO) of developing their own computing infrastructures [2,3]. Focusing on IaaS clouds where users reserve computing resources represented as virtual machines (VMs), storage space and network bandwidth for building up their own computing infrastructures, we study the problem of cloud resource management. On one hand, users have to pay providers for cloud resource usage cost based on the pay-per-use model. On the other hand, they need to efficiently manage such resources to achieve the performance of their applications and improve the utilization of reserved resources, thereby minimizing the usage cost. Reserving large amount of cloud resources may result in good performance of the applications but the resource usage cost will dominate. Thus, intelligent resource management mechanism is needed to efficiently exploit a sufficient amount of cloud resources while satisfying the performance requirement of the applications.

Many complex and large-scale applications have been migrated to clouds for running and processing big amount of data. Examples include surveillance systems, health care systems, smart home or environment monitor [34]. Such applications generate a huge volume of data every day at varying rate from different geographical

* Corresponding author.
E-mail addresses: kang_seungmin@u.nus.edu (S. Kang), elebv@nus.edu.sg (B. Veeravalli), mi_mi_aung@dsi.a-star.edu.sg (K.M.M. Aung).

https://doi.org/10.1016/j.jpdc.2017.10.006
0743-7315/

locations. To handle such high volume and complex data, current trend is towards adopting an infrastructure that spans across multiple clouds and data centers, referred to as *Cloud-of-Clouds* or *multi-cloud* platforms. The users request cloud resources from commercial clouds in different regions corresponding to different data sources and build up specific computing platforms for their applications. Although the adoption of federated multi-clouds can address requirements for the complexity and large scale of applications, efficiently managing the resources in such platforms becomes challenging due to the dynamic arrival of data from different sources. This paper aims at providing users a novel approach for resource management for big data processing in multi-cloud systems.

Generally, we can refer to the amount of work to process data on a computing node as a *load*. For any realistic cloud computing platform, the number of loads submitted to the system is usually much larger than the number of available computing nodes. The problem of load assignment to computing nodes is therefore the most important and challenging since it directly affects the system performance. However, designing an efficient assignment strategy, which is known as *scheduling strategy*, faces many challenges that will be listed below. The main focus of this paper is how to schedule dynamically arriving loads across multi-cloud platforms to improve the performance by considering characteristics of the infrastructure components that a scheduling strategy needs to take into account.

Load balancing. An efficient scheduling strategy should guarantee the load balancing among computing nodes. This will avoid the scenario that a node may have to process many loads in a long period while other nodes are idle. Such scenario prevents users from achieving high resource utilization and degrading the performance. Users may need to pay a higher cost due to longer resource usage time when the running time of application is prolonged. The problem will be more difficult when we consider a more complex system architecture that includes multiple load sources, e.g., a monitoring system with many sensors that capture the data and send to different storage servers for processing and storing. In such a system, the pool of computing resources might be shared among load sources to minimize the usage cost, making the problem of load balancing harder.

Availability and heterogeneity of nodes. Computing nodes, i.e., the VMs requested from clouds, are frequently unavailable due to unexpected failures or they are shutdown for maintenance. Knowing the moment when nodes are available, known as *release time*, is important to the scheduling strategy since loads can be assigned only to released nodes. With a priori known release times, the scheduling strategy can statically assign loads to computing nodes well ahead before the actual load processing. However, with unknown release times, the scheduling strategy needs to react dynamically to new released nodes. Furthermore, if it is assumed that nodes are able to process only one load at a time, the scheduling strategy then needs to be aware of the *ready time* of nodes, i.e., the moment when nodes finish the processing of already received loads. Among released nodes, the node with the earliest ready time is then selected for a new arriving load to achieve load balance.

The ready time of nodes depends on the processing time of loads, which in turn depends on the capacity of computing nodes and the nature of loads. Since nodes are usually heterogeneous in terms of computing capacity, the processing time of a load may vary on different nodes. Furthermore, due to the nature of loads submitted to the system, loads may require different amount of computations. For instance, one load deals with compression and the other deals with encryption using different key lengths, resulting in different amount of running time. Throughout this paper, we refer to this characteristic simply as *computation requirement*.

Network topology and link capacities. Most of existing scheduling strategies do not consider a specific network topology [4,24]. They assume that the data transmission is performed over the Internet. The total processing time of applications including the data transmission time and execution time will be interfered by other users who are using the Internet. While the interference is tolerable for some applications, many applications are requiring a stable bandwidth on the links they are using, e.g., Content Delivery Networks. Such applications have been deployed on a dedicated platform with specific network topology and link capacities to guarantee the Quality of Service. With the development of virtualization technologies, such applications are now migrated to the cloud. For instance, Netflix,[1] which is a major online video streaming service provider in North America, moved its data storage system, streaming servers, encoding engine, and other major modules to Amazon Web Services (AWS) in 2010 [1,6]. While network bandwidth and topology are big issues in a grid computing environment [8], in a multi-cloud environment, the users can easily deploy different network topologies. However, the usage cost of bandwidth prevents users from requesting full bandwidth for the links. Thus, considering the network bandwidth and topology in the scheduling problem is challenging since it directly affects the system performance and resource utilization.

Although many previous studies have considered the scheduling problem [4,24,32], they do not address all the challenges mentioned above. In this paper, we present a novel scheduling strategy, namely *Dynamic Scheduling Strategy* (DSS), which achieves high load balancing among computing nodes as well as among load sources. It considers the availability and heterogeneity of computing nodes, network topology and link capacities to achieve high performance, i.e., minimizing the total processing time of all loads submitted to the system.

To achieve a better load balance, beyond the existing approaches, which consider the computing capacity of nodes, the size of loads and the computation requirement of loads, we further apply the *Divisible Load Theory* (DLT), which assumes that loads can be perfectly divided into a number of chunks with different sizes [5]. Indeed, clouds have become an attractive solution for processing divisible loads that come from many streaming data applications such as monitoring systems, continuous write applications as shown in [34]. We implement DLT model by adopting the phase-based scheduling approach [17] that divides the processing of a load into multiple phases. In each phase, a load chunk will be processed on a node. Multiple load chunks of a load can be processed in parallel on different available nodes.

To compute the ready time of a computing node when it is processing other loads, we apply existing prediction techniques, which allow the computing node to estimate the processing time of a certain load chunk based on the historical processing information, i.e., we assume that there exists on each computing node a training dataset that will be used for prediction model. In a realistic scenario, the size of this training dataset may be small at the beginning, but it will be enriched over time along with the arrival and processing of loads.

In summary, the main contributions of our paper are:

- We propose a novel architecture of a multi-cloud system that can satisfy the complex requirements of users on computing resources, network topology and guaranteed quality of services.
- We design a novel scheduling strategy employing DLT paradigm that addresses all challenges and requirements of an efficient scheduling strategy. This is an important contribution to the design of distributed schedulers for multi-cloud platforms in handling large volume of data.

[1] Netflix: www.netflix.com.