Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A Resource Co-Allocation method for load-balance scheduling over big data platforms

Wanchun Dou[a,*], Xiaolong Xu[b], Xiang Liu[a], Laurence T. Yang[c,d], Yiping Wen[a]

[a] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[b] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China
[c] Department of Computer Science, St. Francis Xavier University, Antigonish, Canada
[d] Cyber-Physical-Social Systems (CPSS) Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

HIGHLIGHTS

- A hierarchical control framework for leveraging tasking scheduling and resource allocation over a big data platform is designed.
- A resource co-allocation method for load-balance scheduling is proposed to efficiently leverage tasking scheduling and resource.
- Experimental evaluations are conducted to evaluate the effectiveness of our proposed method.

ARTICLE INFO

Article history:
Received 1 December 2016
Received in revised form 30 March 2017
Accepted 2 July 2017
Available online xxxx

Keywords:
Cloud computing
Big data
Load balance
Resource allocation

ABSTRACT

Recently, an increasingly large number of high-performance computing (HPC) applications have been developed and hosted in clouds. The cloud computing scheme provides a model of utility computing, which is often implemented in the form of big data centers. As HPC applications are usually resource-intensive, how to exploit economies of scale to harmonize the resource allocation among large-scale HPC applications under performance constraints often poses a significant challenge on the load-balance scheduling policies in big data platforms. In view of this challenge, in this paper we propose a Resource C o-Allocation method, named RCA, for load-balance scheduling. Technically, the method is promoted by four steps that are enacted in a hierarchical control framework. The effectiveness and efficiency of the proposed method are validated by extensive experiments.

1. Introduction

The cloud computing scheme provides a model of utility computing, in which a provider makes computing resources and infrastructure management available to the customer as demanded and charges on the actual resource consumption [1]. Big data centers make the idea of cloud computing into life [2]. Concretely, cloud computing provisions infrastructure resources in the way similar to the delivery of traditional utilities like electricity, water, and gas, and attempts to reduce the overall costs by assembling large pools of computing resources into big data centers. Taking advantage of the latest computing and communication technologies, these big data centers exploit economies of scale, and reduce the overhead of delivering services [3,4]. They provide an appealing computing environment to a large user population. With the popularity of cloud computing, the computing resources are often provisioned in form of cloud services [5]. Over the last decade cloud service providers (CSPs) such as Amazon, Google, and Microsoft, have built their own big data centers running over one million servers each.

Through virtualization technology, cloud users could share data center resources and services, instead of setting up their own infrastructure [6]. In this situation, real-time co-allocation in a load-balancing way plays a key role in the computing resource and the large-scale applications. It signifies the decisions of processing power (or CPU time), memory, storage, network bandwidth etc., among different service requesting [7–9]. For delivering to cloud users fast and flexible provisioning of resources, effective resource management and allocation are indispensable, which makes it possible to offer on-demand network access to configurable computing resources over a big data platform [10].

As a result, resource management in large-scale applications over a big data center poses significant challenges and generates multiple research opportunities. For example, during computing and communication of a large-scale application among high-performance computing (HPC) clouds promoted by a big

* Corresponding author.
E-mail address: douwc@nju.edu.cn (W. Dou).

http://dx.doi.org/10.1016/j.future.2017.07.009
0167-739X/© 2017 Elsevier B.V. All rights reserved.

data platform, the average server utilization is low, while the power consumption of clouds based on over provisioning is excessive [6,9,11]. It leads to a negative ecological impact. As we live in a world with limited resources, over provisioning is not sustainable either economically or environmentally. Therefore, as mentioned in [4], successful resource management not only great benefit big data platform in electrical utility cost saving, but also guarantee fast and flexible provisioning of resources with the freedom from long-term investments.

In view of the observation that big data centers has matured and established itself as an indispensable part of information technology in recent years [2]. Resource management and task scheduling play an essential role, especially when taking the possible convergence of big data analytics and high-performance computing into consideration among large-scale applications that access a big data center in a concurrent way [12].

Generally, new strategies, policies and mechanisms in resource management aim at satisfying a broader range of application requirements, such as allowing cloud servers to operate more efficiently [8], reducing costs for the Cloud Service Providers (CSPs) [2], providing an even more transparent environment for cloud users [4], and support some form of interoperability in a scalable and robust way [7]. For satisfying the challenges, in [1], the policies for cloud resource management is grouped into five level for reflecting various management aspects. More specifically, they highlighted the resource management policy from five aspects: (1) admission control; (2) capacity allocation; (3) load balancing; (4) energy optimization; and (5) quality of service (QoS) guarantees. It reflected a broader range of application requirements based on a hierarchical control model. Moreover, in [1], the authors believed that resource management policies should be promoted by disciplined, rather than ad hoc methods. With the goal in mind, four disciplined mechanisms are concluded for the implementation of resource management, i.e., control theory [13,14], machine learning [15], utility-based [16], and economic models [2,17].

Technically, cloud computing paradigm provides an alternative resource pool for various applications running over networks. It is an ideal abstraction that could provide nearly unlimited computing resources in an elastic way [2,4,6,18]. The cloud platforms are being increasingly considered for HPC applications, which have traditionally targeted grids and supercomputing clusters [18,19]. Generally, the HPC applications needs a large amount of resources, which could be responded by HPC clouds. As the HPC applications are often data-intensive, to process the HPC applications efficiently, it is necessary to deploy the big data platforms in HPC cloud infrastructure for big data analytics. Besides, benefit from the HPC cloud infrastructure, the big data platforms can also support efficient big data storage and access. Typically, HPC applications are resource-intensive scientific workflow (in terms of data, computation, and communication) [18]. In the past few years, more and more HPC applications are developed and hosted in the cloud [18–21] with certain performance constraints. With this tendency, how to exploit economies of scale to harmonize the resource allocation among large-scale HPC applications with performance constraints often challenges the load-balance scheduling policy over big data platforms.

With these scenarios, in this paper, a Resource C o-Allocation method named RCA is proposed for load-balance scheduling over big data platforms. Our contributions are three folds. Firstly, formal concepts and formalized load balance analysis are presented to investigate the RCA method. Secondly, a corresponding method is designed for load-balance scheduling enabled by four step, which could efficiently leverage tasking scheduling and resource allocation over a big data platform, especially HPC clouds. Finally, comprehensive experiments and simulations are conducted to demonstrate the validity of our proposed method.

Table 1
Key terms and descriptions.

Terms	Description
N	The number of meta services
M	The set of meta services, $M = \{m_1, m_2, \dots, m_N\}$
W	The number of cloud services
S	The set of cloud services, $S = \{s_1, s_2, \dots, s_W\}$
m_n	The nth $(1 \le n \le N)$ meta service in M
s_w	The wth $(1 \le w \le W)$ cloud service in S
Q	The type value of the virtualized resources
R	The resource type set, $R = \{r_1, r_2, \dots, r_Q\}$
r_q	The qth $(1 \le q \le Q)$ type resource in R
$a_{n,q}$	The required amount of the resource with type r_q
$h_{w,q}$	The resource capacity of the resource with type r_q
$u_{w,q}$	The resource utilization for the qth resource type of s_w
u_w	The overall resource utilization of s_w
K	The number of employed cloud services in S
U	The average resource utilization of employed cloud services
v_w	The variance of the resource utilization for s_w
V	The average value of the variances for all the cloud services

The remainder of this paper is organized as follows. In Section 2, some preliminary knowledge and formal concepts are presented. In Section 3, a resource co-allocation method for load-balance scheduling is proposed. In , experimental evaluations are conducted to evaluate the effectiveness of our proposed method. Related work and comparison analysis are described in . Finally, in conclusion is presented and future work is discussed.

2. Preliminary knowledge

To facilitate resource co-allocation for load-balance scheduling over big data platforms, formal concepts and load balance analysis are presented in this section. Key terms and descriptions used in this section are listed in Table 1.

Generally, the HPC applications can obtain the resource convenience from big data platforms and cloud services. There are a large number of computing tasks contained in the HPC applications. The resource requirements of these tasks are responded by the cloud services. Here meta services are presented to address such resource requirements.

Definition 1 (*Meta Service*)**.** The meta services reflect the service requirements of the tasks for the submitted HPC applications, including resource requirements and time requirements, denoted as $M = \{m_1, m_2, \dots, m_N\}$, where N is the number of meta services.

The meta services get the resources from the cloud services over big data platforms. Let $S = \{s_1, s_2, \dots, s_W\}$ be the cloud services for performing the meta services, where W is the number of cloud services contained in S.

The cloud services are always equipped with a variety of virtualized resources from big data platforms, including processing power (or CPU time), memory, storage, network bandwidth, etc.

Definition 2 (*Muti-type Virtualized Resources*)**.** There are multiple types of the virtualized resources (i.e., CPU, memory, storage, bandwidth, etc.) provisioned by the cloud services, denoted as $R = \{r_1, r_2, \dots, r_Q\}$, where Q is the number of resource types.

Definition 3 (*Resource Requirement of m_n*)**.** The resource requirement of m_n $(1 \le n \le N)$ is the amount of the resources for each type of resources that requested by m_n, denoted as $a_n = \{a_{n,q} | 1 \le q \le Q\}$, where $a_{n,q}$ is the required amount of the resource with type r_q.

Definition 4 (*Resource Capacity of s_w*)**.** The resource capacity of s_w $(1 \le w \le W)$ is the total amount for each type of the available resources, denoted as $h_w = \{h_{w,q} | 1 \le q \le Q\}$, where $h_{w,q}$ is the resource capacity of the resource with type r_q.