Sustainable Computing: Informatics and Systems xxx (2018) xxx–xxx

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

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

Fast resource scaling in elastic clusters with an agile method for demand estimation

Cheng Hu [a], Yuhui Deng [a,b,*]

[a] Department of Computer Science, Jinan University, Guangzhou 510632, China
[b] State Key Laboratory of Computer Architecture, Institute of Computing, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 19 December 2017
Received in revised form 31 January 2018
Accepted 5 March 2018
Available online xxx

Keywords:
Green computing
Elastic cluster
Demand estimation
Resource scaling
Resource management

ABSTRACT

For energy saving, elastic clusters are introduced to cut back the energy wasted on powering unused servers. In an elastic cluster, the number of working servers, or called resources, is dynamically scaled based on resource demand of workload. However, many traditional scaling methods are unaware of an exact resource demand of workload. They gradually scale resources according to current service level with loose demand estimations or even with no estimation. Additionally, to provide the ability to make precise demand estimations, some other methods are proposed. They artificially represent system situation with a general model, but the model may not well reflect the reality because it is often difficult to describe the real situation of a system. For both of these methods, resources cannot be exactly scaled to the demand when demand changes, and there is a time delay before resources are scaled to the demand. This scaling delay will incur a performance degradation when workload increase, and will cause an energy waste when workload decrease. In this paper, we strive to efficiently estimate the actual demand of workload and achieve fast resource scaling in elastic clusters. Unlike traditional methods which make great efforts to understand the complex system situation, we only concentrate on the information of past actual resource demands. This information is actually the most straightforward and valid reflection to the real situation of a specific system, so it contains valuable knowledge for estimating the actual resource demand of new incoming workload. Therefore, we propose an agile method to directly estimate resource demand based on that knowledge, thus achieving a high accuracy. Specifically, our method directly learns that knowledge through a learning method—random forests, so it does not need artificial system analyses which are both complex and time-consuming. In addition, it is efficient to build random forests and make resource estimations in our method. Thus, our method can be efficiently and agilely performed in elastic clusters to reduce the scaling delay and achieve fast resource scaling.

1. Introduction

With the rapid advances in information and communication technology (ICT) such as global interconnection of heterogeneous information systems, tremendous structured and unstructured data are generated from a wide range and multiple data sources [1,2]; for instance, numerous data of video and pictures can be produced by large-scale video surveillance systems [3]. As a result, to maintain these tremendous data and provide corresponding services, a large requirement on both capacity and scale is presented to data centers (DCs). For example, mobile cloud computing is an application of DCs to maintain and process a big volume of data [4]. The overhead for a DC powering all its servers is huge, however, a small part of servers can most of the time fulfil the demand of workload. The reasons are that DCs are designed to be capable of handling peak workload, but workload is usually at a low level. In most DCs, average server utilization can be as low as 20%, and powering idle servers consumes as much as 60% the overall power [5].

Elastic clusters are introduced to DCs, in order to cut back the energy wasted on powering idle servers. In such clusters, the number of working servers, or called resources, are scaled on demand. Besides, the other unused servers are turned off or switched into a low power state, for energy saving. Generally, system performance can be evaluated by the average scheduling delay of user

* Corresponding author at: Department of Computer Science, Jinan University, Guangzhou 510632, China.
E-mail addresses: hucheng_public@163.com (C. Hu), tyhdeng@jnu.edu.cn (Y. Deng).

https://doi.org/10.1016/j.suscom.2018.03.001
2210-5379/

requests, and service-level agreement (SLA) is negotiated so as to fix the maximum average scheduling delay. To meet the performance objective in terms of task (request) scheduling delay, two ways are widely used to scale the resources and match demands. One way is to directly observe current service level, and scale resources if needed [6–8]. The other way is to estimate the actual demand based on some workload features (service rate in general [9]), and then scale resources based on the actual conditions of workload [10–12,9]. For the first way, the actual amount of resources which can fulfil current workload is unknown. To achieve on-demand provision, it has to gradually scale until meet the SLA. In addition, if demand changes, the change cannot be detected immediately, because the service level will stay at the previous level for a while. Therefore, there is a time delay before resources are scaled to the actual demand. For the second way, some time is taken to evaluate workload features (conditions) and estimate the actual resource demand. So, when resource demand changes, a time delay is needed to detect it. Besides, the resource demand of workload is always hard to estimate, and is affected by many factors. Even the arrival rata of requests is the same, due to different service time, skew inter-arrival time and some other factors, the service rate of a server varies. As a result, different number of working service are required to maintain a same service level. Hence, resources cannot be accurately scale to the actual demand, a follow-up adjustment is needed. Therefore, not only in the first way, the second way also brings a delay before resources are scaled to the actual demand. A SLA violation can bring great harm to service providers [13]. For example, an extra latency of 500 ms on search page display can cut back revenues of Google by 20%.

When workload increase and more servers are required, the time delay (or called scaling delay) mentioned above will incur a performance degradation. On the contrary, when workload decrease and less servers are enough, the scaling delay will lead to an energy waste. Therefore, reducing the scaling delay and achieving fast resource scaling can further promote system performance and cut back energy consumption. According to the above paragraph, to reduce the scale delay: first, the actual demand should be estimated as accurately as possible; second, demand changes should be detected as soon as possible. To make an accurate estimation on actual resource demand of workload, many mathematical approaches are proposed by artificially analyzing and modeling the relations between actual resource demand and some workload features (e.g., request arrival rate, and service time [14]) and then make estimation based on actual workload features. But the relations are obscure and hard to analyze. For example, considering a cluster as a queueing service system, queueing theory [15] reveals that the relations vary a lot due to various queueing models. Therefore, creating model instances for a given system can be a complex and time-consuming task [14]. Besides, most analyses contain some constraints in order to obtain coherent conclusions. However, due to many unpredictable conditions in real systems, the real situation is intangible and probably out of analyzers' settings. Artificial analyses can hardly describe the real situation of a system. What is more, even these mathematical approaches can make an accurate estimation, they are complex and time-consuming processes, and cannot detect demand changes in time.

In this paper, we present an agile method to estimate the actual resource demand of workload in an elastic cluster whose SLA is about average request scheduling delay. This method leveraging a learning algorithm—random forests to directly make an accurate estimation based on observed workload features. Unlike traditional methods which make great efforts to understand the complex system situation, our method only concentrate on the information of past actual resource demands. This information is actually the most straightforward and valid reflection to the real situation of a specific system. With the help of random forests, the knowledge of resource demands under diverse workload features can be efficiently extracted from these information, and no artificial analysis on relationships between workload and demand is needed. According to that knowledge, our method truly makes an accurate estimation which is fit to the real situation, and eliminates the risk of deviating from reality. As a result, our method can effectively reduce the scaling delay and achieve fast resource scaling. The main contributions of this paper are as follows:

1. With the aid of queueing theory, we use a general system model to represent an elastic cluster. According to the system model, we show the fundamental reason for researches making theoretical analyses to understand system situations is that they consider the serving process of a cluster to be orderly. However, we reveal that the reality is something beyond researches' settings, and it is a big challenge to achieve a good performance on resource estimation.
2. We propose a resource demand estimating method, which can directly make an accurate estimation based on workload features, to achieve fast resource scaling. Different from some other methods, our method concentrates on the information of past actual resource demands, and avoid the difficulties for understanding the complex situations of clusters. Moreover, by leveraging random forests, our method can efficiently extract the knowledge of resource demands according to past demand information.
3. We perform extensive experiments with real workload traces to evaluate our method along with other three representative ones. We evaluate the effectiveness of our method from two aspects that are estimation accuracy and agility for resource scaling. Besides, we discuss the time overhead for our method to perform resource scaling in elastic clusters, and validate the effectiveness of our method.

The rest of this paper is structured as follows. Section 2 discusses the related work. Section 3 introduces a general system architecture for modern elastic clusters. Section 4 analyses the resource demands of workload in elastic clusters and presents our resource demand estimating method. Section 5 performs comprehensive experiments to evaluate our method along with other three representative ones. A discussion of the work is given in Section 6. Section 7 concludes the paper.

2. Related work

The existing researches on constructing elastic clusters for energy saving can be mainly classified into two categories, according to the methods they are used for scaling resources.

In the first category, resource scaling is made by a reactive way. The reactive way here means, when a mismatch between resources and workload is detected, resource scaling is made as a reaction. Meisner et al. [6] build an elastic multi-service system, in which an energy-conservation approach—PowerNap is used to realize resource scaling. Dynamic voltage and frequency scaling (DVFS) is adopted in the system, in order to realize rapidly transition the state of a server between an active state (high-performance) and a nap state (minimal-power). In the system, idle servers are transitioned to the nap state for energy saving. When request arrivals are detected by the network interface card (NIC) of a nap server, the server is woken up for service. After all requests are finished, the server are transitioned to the nap state again. Although DVFS can achieve fast state transitions, it only applicable to a few components of a server (such as CPU). Thus, for outstanding efficiency of energy saving, state transition of a whole server is more widespread. Entrialgo et al. [7] propose a power management technique to realize

Please cite this article in press as: C. Hu, Y. Deng, Fast resource scaling in elastic clusters with an agile method for demand estimation, Sustain. Comput.: Inform. Syst. (2018), https://doi.org/10.1016/j.suscom.2018.03.001