Future Generation Computer Systems 58 (2016) 42–55

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Optimization of virtual resource management for cloud applications to cope with traffic burst

Qizhi Zhang, Haopeng Chen *, Yuxi Shen, Sixiang Ma, Heng Lu

School of Software, Shanghai Jiao Tong University, Shanghai, China

HIGHLIGHTS

- We present a framework of dynamic resource management to cope with traffic burst.
- The prediction of traffic burst is based on Gompertz Curve and Moving Average model.
- VM scheduler involves VM Provisioning, VM Placement and VM Recycling.
- High availability and cost-effectiveness are achieved by the proposed framework.

ARTICLE INFO

Article history:
Received 18 May 2015
Received in revised form
28 November 2015
Accepted 18 December 2015
Available online 6 January 2016

Keywords:
Cloud computing
Workload prediction
Virtual resource management
Traffic burst

ABSTRACT

Being the latest computing paradigm, cloud computing has proliferated as many IT giants started to deliver resources as services. Thus application providers are free from the burden of the low-level implementation and system administration. Meanwhile, the fact that we are in an era of information explosion brings certain challenges. Some websites may encounter a sharp rising workload due to some unexpected social concerns, which make these websites unavailable or even fail to provide services in time. Currently, a post-action method based on human experience and system alarm is widely used to handle this scenario in industry, which has shortcomings like reaction delay. In our paper, we want to solve this problem by deploying such websites on cloud, and use features of the cloud to tackle it. We present a framework of dynamic virtual resource management in clouds, to cope with traffic burst that applications might encounter. The framework implements a whole work-flow from prediction of the sharp rising workload to a customized resource management module which guarantees the high availability of web applications and cost-effectiveness of the cloud service providers. Our experiments show the accuracy of our workload forecasting method by comparing it with other methods. The 1998 World Cup workload dataset used in our experiment reveals the applicability of our model in the specific scenarios of traffic burst. Also, a simulation-based experiment is designed to indicate that the proposed management framework detects changes in workload intensity that occur over time and allocates multiple virtualized IT resources accordingly to achieve high availability and cost-effective targets.

1. Introduction

During the past few decades, cloud computing [1,2] has been one of the hottest topics in computer science. Cloud computing allows users to deploy applications in the cloud data center and thus saves them from purchasing physical infrastructures and some maintenance cost. As a new computing model, cloud computing has led to several revolutions. Instead of buying infrastructures, people start to deploy their applications in public data centers, which we call clouds. With more and more data centers being set up, virtual resources are delivered as services by more and more IT giants. Furthermore, facilitated by the development of virtualization technology, such as Xen [3] and KVM [4], a lot of IaaS (Infrastructure as a Service) platforms, like Openstack,[1] Amazon EC2,[2] have been able to provide mature services, which make them more and more important in the industry. Cloud services promise scalability, cost-effectiveness and availability, which attract an increasing number of users to migrate their web applications to a

* Corresponding author.
E-mail address: chen-hp@sjtu.edu.cn (H. Chen).

http://dx.doi.org/10.1016/j.future.2015.12.011
0167-739X/

[1] http://www.openstack.org/.
[2] https://aws.amazon.com/ec2/.

cloud environment. With the increasing use of cloud data centers, more and more researches start to focus on problems in clouds.

A lot of researches have put forward several resource management frameworks to achieve various goals in data centers. Nguyen Van et al. [5] resort this problem to a Constraint Programming approach to formulate and solve the optimization problem. Beloglazov et al. [6] put their attention on energy saving, and propose a framework which can save a great deal of power by the scheduler of virtual machines. Whats more, Sotomayor et al. [7] even implement all their designs into a mature project, and make it open source for public research. All these researches are trying to find a best framework that can solve problem in general. However, the multi-targets property of such problems makes it difficult to figure out a framework that can fit all the constraints. A "perfect" framework like this may require multiple iterations to satisfy all the constraints, let alone possible conflicts among different constraints. But to application providers, not all these constraints are of the same importance, some care about response time while others care more about availability. So we need to customize solutions for special scenarios in many cases. For example, M. Li et al. [8] propose an innovative resource scheduler named CAM, which is particularly designed for data center clouds hosting MapReduce jobs.

The information explosion issue is another concern in this era of speeding development. Especially with the fast proliferation of social networks, people are more interacted than ever before and hot topics are propagated much faster and more widely. Thus some topics may suddenly become the focus of public opinion from worldwide, like hot topics on twitter, Snowden event, "I paid a bribe" website in Indian.[3] In these cases, it is very likely that the too much attention on those topics will bring huge workload to related websites, even result in failures. However a small ratio of access failures would cause user dis-satisfactions, and may further results in a loss of potential users. More seriously, it may eventually lead to a substantial financial impact such as loss of business opportunities and customer dissatisfaction. Currently, a post-action method based on human experience and system alarm is now widely used to handle this problem in industry. Unfortunately, a time-latency is unavoidable adopting these methods, which fails to foresee the sudden workload change and make quick solutions.

To deal with the above challenges, we present a customized resource management framework, which makes predictions, detects traffic bursts and provisions virtual resources to promise high availability as well as cost effectiveness. Our paper addresses the following problems:

(1) A novel prediction model to handle traffic bursts and provide accurate predictions of real time workloads.

(2) A customized resource management framework to tackle the problem of traffic burst, guarantying high availability and achieve the goal of cost-effectiveness as well.

The rest of our paper is organized as follows. Section 2 illustrates the architecture of our framework and Section 3 focuses on the prediction model we propose. Section 4 describes the performance monitor. Section 5 describes the virtual machine scheduling strategy and the corresponding experiment details are illustrated in Section 6. Section 7 presents the related work and Section 8 discusses some conclusions with some future work.

2. System architecture

The architecture of our resource management framework and some data flows are illustrated in Fig. 1. Our system mainly consists of three main components, the Workload Forecasting Module, the VM Scheduler and the Performance Monitor.

The Workload Forecasting Module keep track of history workloads, detects possible workload change and make predictions of future workload, which will be used to provision virtual resources in the next time period. In this way, virtual resources are timely provisioned according to real time workload, which will prevent request failures resulted from a resource shortage.

The VM Scheduler resolves how much resource to provision under the predicted workload so that the performance of the application will remain in a level that will not dissatisfy the users. Then it will also place these resources (VM instances) to different zones or regions to ensure a high availability of that application. Considering a possible declining workload, a Resource Collector is designed to recycle resources when necessary.

The Performance Monitor is responsible for collecting runtime performance data of applications, virtual machines and physical machines. And it is used to determine whether the application is provisioned enough virtual resources. It runs periodically to avoid bringing too much burden to the hosts.

Apart from our framework, Fig. 1 also presents the cloud environment. The cloud data center consists of a large number of physical machines (PM). Each set of VMs are called a region, and several regions are united as a zone. Each PM hosts multiple VMs through a hyper-visor. Cloud applications are deployed on a number of VMs as different instances. Application users can send requests to the cloud and may potentially cause traffic bursts.

The basic functions of the three main components of our framework are discussed as follows.

2.1. Workload forecasting module

Workload Forecasting is a topic that has been discussed a lot and there are already some proposed models to deal with a normal workload forecasting scenario. However, in this era of information explosion, sharp workload rises are becoming more and more common and frequent. These classic forecasting models are not fast enough under these circumstances, which may result in a prediction delay and provision insufficiency.

Here we provide a workload forecasting model to deal with traffic burst. This model is responsible for generating predictions of the future workload based on the former request pattern of the application. Such predictions will be used as input for the VM scheduler model. Users are also allowed to configure parameters in this model to suit for their own applications.

2.2. Performance monitor

To ensure performance of cloud applications as much as possible, a performance monitor is necessary to keep aware of the application performance. In our framework, this model collects and tracks the data of resource utilization on various metrics of VMs, which also acts as an important input for the VM scheduler model.

Monitoring data can be obtained via regular monitoring tools or by Cloud monitoring services such as Amazon Cloud-Watch.[4] And in the later simulation experiment, we obtained the performance data via the simulation platform API.

This module runs periodically with a low priority, collects resource usage data including CPU, memory and I/O bandwidth of all virtual machines and physical machines.

[3] http://www.ipaidabribe.com/.

[4] http://aws.amazon.com/cloudwatch/.