Self-adaptive Resource Management For Cloud Computing Environments

Frank Mensah (6830725)
Department of Computer Science
University of Paderborn

July 15, 2018

Abstract

Although, cloud computing is increasingly gaining fame in most enterprises, there are issues known with respect to the efficient resource allocation and management of virtual machines (VM) within the cloud computing environment. The efficient partitioning of the hardware resources into VMs promotes flexible computing environment. However, problems such as difference in time demands of users and the arbitrary deployment of multi-tier applications makes it difficult to properly configure a VM. We employ Self-adaptive management systems which uses an efficient reinforcement learning mechanism for managing the virtual machine resources and also a Resource Allocation Optimizer which uses a Support Vector Regression (SVR) which seeks to satisfy the Service Level Agreements (SLA) requested by customers to address the challenges of VM configuration. We handle the cloud resource provisioning as a distributed learning mechanism with an efficient learning algorithm with the introduction of iBalloon prototype system. We show iBalloon as a scalable VM resource allocator by its application of a Decision-maker component which learns intelligently to configure the VM in order respond to all changing demands of applications on-the-fly.

1 Introduction

There is an increasing demands for computing power in this digital era as humans rely profoundly on computers in order to achieve numerous tasks.This demand for computing power is pushing a lot of studies in technologies like the cloud computing and virtualization. Simply put, cloud computing is the delivery of computing services like servers, storage, databases, networking, software, analytics and more over the Internet (the cloud). Companies offering these computing services are called cloud providers and typically charge for cloud computing services based on usage, similar to how water or electricity is billed at home [4]. Cloud computing provide applications as services over the Internet and also hardware and systems software services in data centers [1].

The hardware and systems software which are provided by cloud computing are commonly known as Infrastructure-as-a-service (IaaS) where as the applications delivered as a service are also commonly known as Software-as-a-Service (SaaS) [6].

To ensure better performance boost of services by the cloud computers, the virtual machines should be able to resolve the challenges faced with the allocation of its host resources. That is, the virtual machines should be able to effectively resize themselves in response to any changes of an application demand and to also handle any arbitrary deployment of multi-tier applications on the cloud system [6]. A key impediment to performance boost in cloud computing is the issue of VM configuration. It is crucial to address the problems in VM configuration by making them autonomous in taking decisions in order to optimize the allocations of their resources to applications on demand. To achieve such levels of autonomy, Reinforcement Learning (RL) processes are incorporated into the operations and decision making of the VM. Reinforcement Learning is the process of learning by interaction with dynamic environment, which generates the optimal policy for a given set of states [5]. This paper seeks to present self adaptive mechanisms for cloud management with the aim of addressing problems in effective resource allocations within a scaled number of virtual machines in a distributed environment [6]. Specifically, we look at these topics, briefly introduced below into details in sections 3 and 5;

- **Self-Adaptive Capacity Management** : In traditional physical servers, the resource is well defined as the server can be identified as an individual entity. When there is a need for more capacity, one can easily upgrade the servers or purchase additional ones. However, in a virtual scope of the cloud computing environment, managing the capacity is much more difficult since there are many infrastructure components working together to attain a common goal. Therefore balancing the resources becomes very crucial. In a scenario where a host machine exhausts its physical memory capacity, The number of VMs it can run gets limited though it may have other resources available. Resources like the CPU and the Storage may all be available and in full capacity despite the limitation of the servers physical memory. The effort in keeping all resources balanced in a virtual environment isn't an easy task and require a capacity management approach which goes beyond normal static configurations as seen in traditional servers. An efficient reinforcement learning approach for the management of the virtual machines capacity is employed to optimize and also to predict the future resource needs of all applications demands in the cloud environment [6].The VM capacity refers to the resources of an individual VM and their right-sizing. The VM resources could be the CPU, memory or the I/O devices required by the VM in its execution in order to meet the current and future business needs in a cost-effective way[6].

- **Resource Allocation Optimization** : In the implementation of a self-adaptive capacity management of the VM, RL Algorithms such as the Application Service prediction module and the Support Vector Regression (SVR) are employed to ensure that the proposed system can satisfy the Service Level Agreement (SLA) requested by a customer [2]. The application SLA, is the contract signed by consumers of cloud services with their service providers in order to determine the

price of each services with consumers. The contracted content sometimes refers to certain performance metrics of resources, such as the performance of the CPU,the memory capacity,and the response time [2]. In general, RL algorithms helps to optimize the resource allocations in the VMs.

2 PROBLEM DEFINITION

In order to better appreciate the efforts that researchers put in their work in finding a solution to self adaptive resource management in cloud computing. We first have to familiarize ourselves with the problems that are commonly known to impede the operations of a cloud system.

- **Issues of VM Reconfiguration** :After an initial configuration of a VM, its resource or capacity management relies precisely on the set configuration values for its operations. However, the effect of reconfiguration cannot be perceived as there are usually up to 10 minutes delay time before a memory reconfiguration stabilizes [6]. Experiments conducted by Rao et al. 2011 indicated the problem at hand such that, the virtual CPU (VCPU) was removed every 5 minutes until one was left at the 15th minutes. The VCPU was later added back one-by-one until the initial capacity was attained. The increase in the VCPUs resulted the capacity of the overall VCPU to move from 1 to 2 but with a 5 minutes delay before the response time stabilized on the 25th minute as illustrated in Figure 1. It was identified that the delay for the VM to realize the changed configuration was due to the backlog of request when there were more CPUs available which caused the VM to spend minutes into digesting the congested requests. The issue of delays are a problem since the 10 minute wait is largely significant in the cloud computing environment and it needs to be addressed. Figure 1 illustrates the experimental results.

- **Cluster Wide Correlation** : In a multi-tier public cloud system, there is a need for proper configuration of the tiers. Tiers usually works synchronously such that a delay in one tier will affect all nodes in the cloud and causing a slow down [6]. Experiments again conducted by Rao et al. 2011 indicated that an increase in the VCPU of the back tiers causes the CPU of the front-end tier to decrease considerably. Reason being that, the faster processing of requests by the back tier saves the front tiers from spending more resources in processing unfinished requests.

  In Summary, the issue of VM reconfiguration should be dealt with in a manner that allows the VM to resize itself in response to any time-varying resources and application demands that come upon it. It should also be able to guarantee VM's application-level performance in the presence of complicated resource to performance relationship [6]. iBalloon system is therefore introduced as a system to implement the individual VMs as a distributed management framework by decoupling the functionalities of the cloud system into three components.

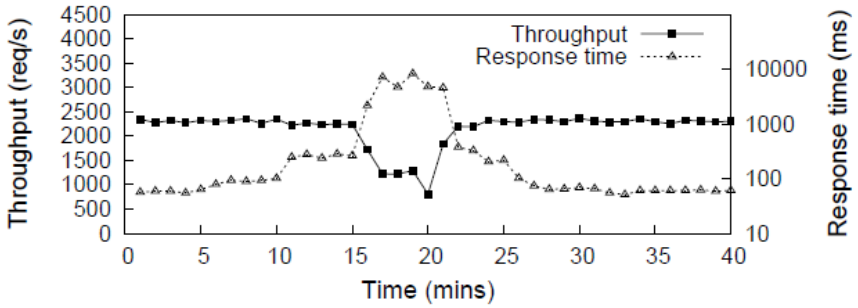

Figure 1: Figure 1 [6] illustrates the delayed effect of removing the VCPUs of a VM in sessions of 5 minutes interval and later adding them back to their original capacity. The response time decreased after 15 minutes of removing the VCPU of the VM in 5 minutes succession. The CPU stabilizes within 10 minutes after adding all the VCPUs back.The delay is mainly caused by the time that it takes for the CPU to digest the pending request and begin processing.

DB VCPU	1VCPU	2VCPU	3VCPU	4VCPU
APP MEM	790MB	600MB	320MB	290MB
APP CPU%	61%	47%	15%	10%

Figure 2: Table 1 [6] shows the effect of increasing the VCPU at the back-end tier as the CPU of the front-end tier decreases.

3 SELF-ADAPTIVE CAPACITY MANAGEMENT WITH IBALLOON

The authors in [3] explained capacity management of a virtual machine as the resources used in order to meet the unique requirements of a cloud computing system. These resources could be the memory, CPU or storage [1]. In the Self-Adaptive Capacity Management, iBalloon is introduced as the design approach [6]. In iBalloon, every individual virtual machine is initialized as a distributed management framework [6]. The Distributed management framework simply refers to the decoupling of functionalities in of the iBalloon into three components: *Host-agent, App-agent and Decision-maker* as shown in the Figure 3.

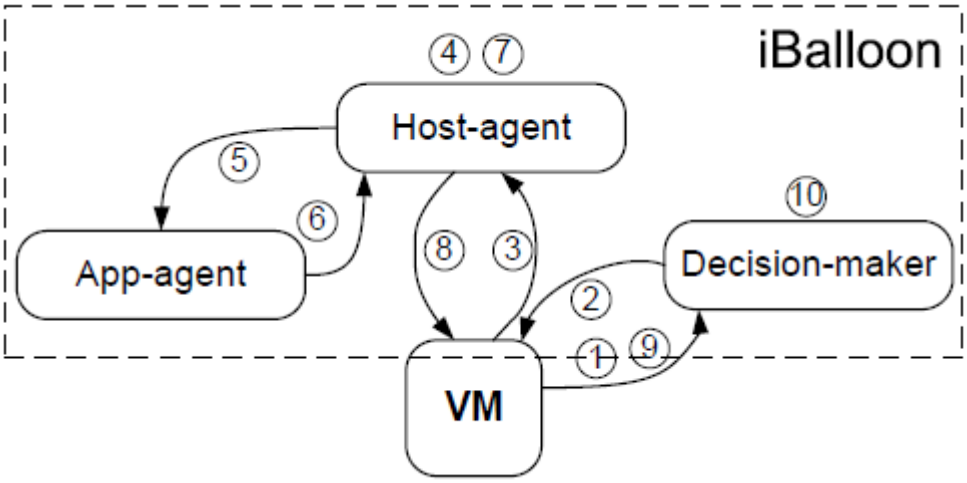

Figure 3: Figure 2 [2] illustrates the iBalloon architecture as a Distributed management framework.

There is a **Host-agent** for each host machine with the responsibility of allocating the hosts hardware resources to the virtual machine and giving a feedback. The **App-agent** is also responsible for maintaining the application SLA profile and reports the run-time application performance. Finally is the **Decision Maker** which serves as the learning agent for each virtual machine in order to realize the automatic capacity management.

So, in the iBalloon architecture, the virtual machine first report its running status, then the Decision Maker replies with a capacity suggestion. The virtual machine then submits the resource request to the Host-agent which also synchronously collects all the virtual machines request, reconfigures the virtual machines resources and sleeps for a management interval. The Host-agent queries the App-agent a virtual machine application-level performance, then also calculates and sends a feedback. The virtual machine then wraps the information about the interactions and reports to the Decision-maker, which then finally updates the capacity management policy for the virtual machines. There are two assumptions made in this work by the authors;

- Capacity decisions are made based on the virtual machines running status.

- Virtual Machines relies on feedback signals.

The virtual machines running status provide insight into the usage of its resource and has a direct impact on the management decisions. Informations like the constrained or over-provisioned resources are insights that can be inferred by the running status [6]. Also, the feedback signal serves as the dispute settler (arbiter) when resources are being contended. So, these assumptions aforementioned, mainly define the virtual machines capacity management tasks as an autonomous learning process within its interactive environment [6].

4 EVALUATION OF THE DESIGN

The evaluation of the setup for this experiment was done based on the answers to the following questions.

- Can the learned policy be reused to control a similar application or a different platform? [6].
- Can iBalloon properly distribute the constrained resource and optimize overall system performance? [6].
- Is iBalloon scalable? [6].

4.1 CAN THE LEARNED POLICY BE REUSED

The distributed management framework (iBalloon) was tested in two variations in order to better understand its performance. One with an initialized management policy built and updated by the Decision-Maker and the other without any initialization policy. In the non-initialized policy setup, the iBalloon managed to keep 90% of the request it received below the SLA response time threshold. This indicated that, iBalloon managed to quickly adapt to good policies and maintained its performance at a stable range even though it started with poor policies [6]. Also, the initialized policy setup, blocked close to 80% of the SLA violated requests. This is a clear indication of a needed good policy initialization in even more complicated environments [6].

4.2 CAN IBALLOON PROPERLY DISTRIBUTE RESOURCES

iBalloon was tested with a reward metric signal which provides strong incentives to the VM to give up unused resources [6]. After 10 hours of running in order to build a substantial trained policy, the iBalloon was able to expand and shrink the CPU and I/O bandwidth resources as workload varied [6]. In the practical sense, an error caused by the autonomous management of the VMs capacity could be very expensive therefore, iBalloons was tested online with the ability to detect the appropriate resources meant for any specific application demands. Thus, iBalloon was able to detect all misconfiguration and reconfigure itself correctly. Figure 4 illustrates the rewards and their associate traffic levels.

4.3 IS IBALLOON SCALABLE

iBalloon was tested on a dedicated CIC200 cluster with two Tiers each and a total of 128 VMs were randomly deployed on 16 nodes with the assumption of no topology [6]. iBalloon was meant to coordinate VMs on different hosts which runs their own resource allocation policy. The results of the setup showed that after running the setup for 10 hours, a manual tweaking of the cluster produced an optimal strategy. In addition to iBalloon, four other strategies like the work-conserving scheme, Adaptive proportional integral and Auto-regressive-moving-average (ARMA) were also experimented to compare [6]. Figure illustrates the results.

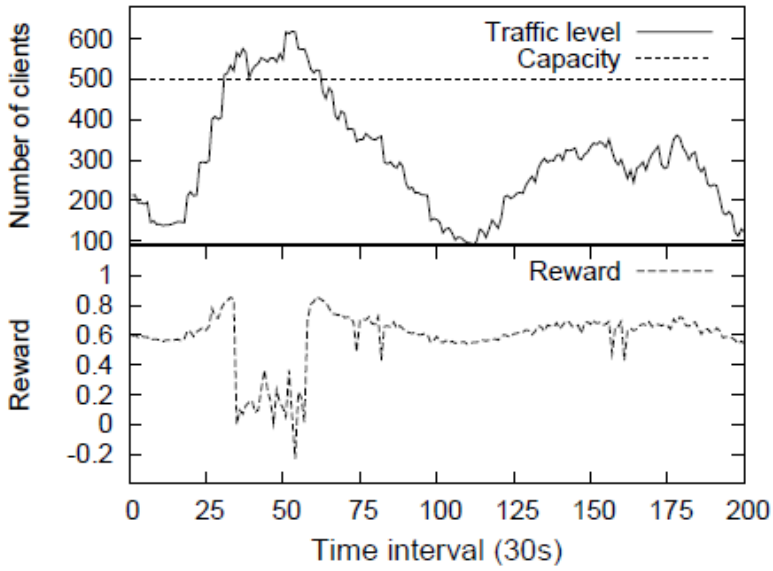

Figure 4: Figure 4 [6] illustrates the rewards with different traffic associated with it.

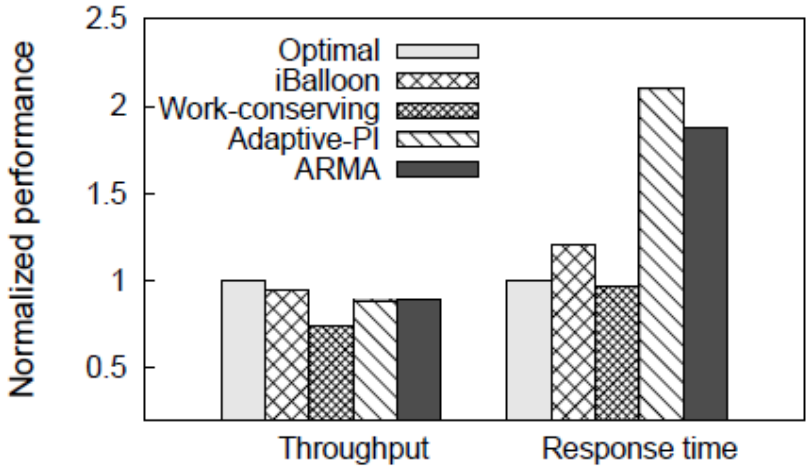

Figure 5: Figure 9 [6] illustrates the various reconfiguration approaches including iBalloon on a cluster of 128 correlated VMs.

5 Resource Allocation Optimization

To optimize the resource allocation in the cloud computing environment, Huang et al. [2013] also illustrates an architecture which employs SVR and a version of an Evolutionary Algorithm (EA) known as Genetic Algorithm (GA) [2]. In the illustration there is an Application Service Resource Pool that is responsible for collecting all the applications which are being given by the Internet service providers. An Application Monitor module then records the overall utilization of the system, whereas a Physical Machine Resource Pool provides the resources to Host CPU, Memory and I/O devices. The lookup and a resource table managed by the physical machine resource pool is used in determining the strategies of increasing and decreasing the number of virtual machines requested by various applications. In oder to ensure adequate satisfaction to the SLA, the Global Resource Allocation Module creates and collects the virtual machines and also use the Generic Algorithm (GA) to redistribute the resources to clients while the SVR estimates the number of resource utilization according to the SLA of each process. Figure 6 shows the architecture of the proposed setup as been illustrated by Figure 1 [2].

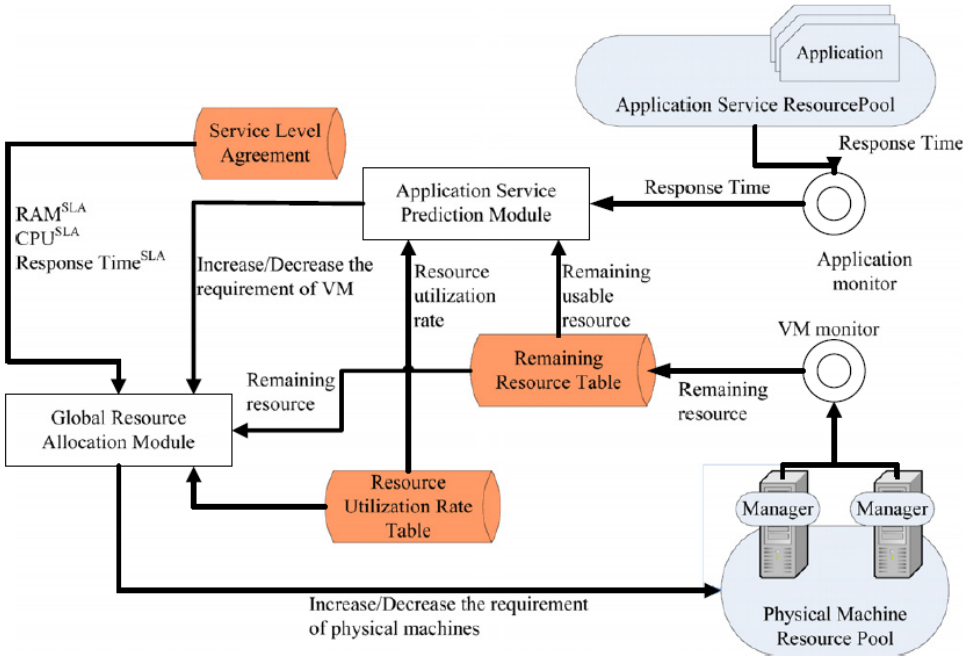

Figure 6: Proposed architecture to optimize resource allocation

6 Conclusion

The implementation of the Self-Adaptive Capacity Management employs a generic framework which makes it possible to incorporate various learning algorithms. The Decision-maker, when implemented with good learning algorithms learns better to implement good SLA policies than having VMs with fixed configuration policies [6]. The proposed approach by Huang et al. [2013], which employs an application service prediction module built with SVR or the Genetic Algorithm (GA) could be implemented within the Decision-maker to optimize its efficiency. It offers opportunities for providing fine grained operations and stable initial performance [2]. In a nutshell, the self-adaptive resource managed cloud system can be achieved by having an efficient and autonomously configured VM with the help of an integrated learning algorithm which can make the Decision-maker of the iBalloon build up good resource allocation policies.

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[2] Chenn-Jung Huang, Chih-Tai Guan, Heng-Ming Chen, Yu-Wu Wang, Shun-Chih Chang, Ching-Yu Li, and Chuan-Hsiang Weng. An adaptive resource management

scheme in cloud computing. *Engineering Applications of Artificial Intelligence*, 26 (1):382–389, 2013.

[3] Sanjay Kumar, Vanish Talwar, Vibhore Kumar, Parthasarathy Ranganathan, and Karsten Schwan. vmanage: loosely coupled platform and virtualization management in data centers. In *Proceedings of the 6th international conference on Autonomic computing*, pages 127–136. ACM, 2009.

[4] Microsoft. What is cloud computing, cloud info, 2018. URL `https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/`.

[5] Jia Rao, Xiangping Bu, Cheng-Zhong Xu, Leyi Wang, and George Yin. Vconf: a reinforcement learning approach to virtual machines auto-configuration. In *Proceedings of the 6th international conference on Autonomic computing*, pages 137–146. ACM, 2009.

[6] Jia Rao, Xiangping Bu, Cheng-Zhong Xu, and Kun Wang. A distributed self-learning approach for elastic provisioning of virtualized cloud resources. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*, pages 45–54. IEEE, 2011.