ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

AN AUDACIOUS RESOURCE MANAGEMENT FRAMEWORK TO ENSURE QoS IN IaaS CLOUD

M.NISHANTHI,

Student, CSE Department,
M.I.E.T Engineering College, Trichy.
nishamageshz@gmail.com

***Abstract* - On-demand dynamic resource provisioning is the major aspiration of the cloud computing. Elasticity is the key feature of the cloud computing. Although elasticity exists, effective virtualized resource management is still one of the most challenging tasks. With the rapid increase in workload, the existing approaches cannot satisfy the growing performance requirements efficiently as it results in an insufficient resource equipping either because of inaccurate adaptation decisions or of the slow process of adjustments. These approaches may degrade high-level QoS and violate Service Level Objective (SLO). To overcome these drawbacks, a generic reinforcement learning-based audacious resource management framework is to be inaugurated to ensure high-level QoS and to limit SLO violation rate for the IaaS cloud, when the workload increases drastically. With this strategy, the resource allocation is substantially increased in each and every adaptation cycle when workload increases. Firstly, this audacious strategy provisions resource which is possibly more than the actual demands, and then reduces the over-provisioned resources if needed. By applying the audacious strategy, high-level of QoS is achieved in the first step with the increase in performance requirement. When compared with the existing efforts, the adaptation time will be speeded up. Thus by enabling quick adaptation, the framework limits the SLO violation rate even with rapidly increasing workload.**

Keywords - Cloud computing, Virtualization, Resource Management, Quality of Service, Service Level Objective.

I. INTRODUCTION

Cloud computing, also known as on-demand computing, is a kind of internet-based computing, where shared resources and information are provided to computers and other devices on-demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. In the most basic cloud-service model and according to the IETF (Internet Engineering Task Force) providers of IaaS offer computers physical or (more often) virtual machines and other resources. IaaS refers to online services that abstract user from the detail of infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc. IaaS clouds often offer additional resources such as a virtual-machine disk-image library, raw block storage, file or object storage, firewalls, load balancers, IP addresses, Virtual Local Area Networks (VLANs), and software bundles. IaaS-cloud providers supply these resources on-demand from their large pools of equipment installed in data centers. For wide-area connectivity, customers can use either the Internet or carrier clouds (dedicated virtual private networks).To deploy their applications, cloud users install operating-system images and their application software on the cloud infrastructure. In this model, the cloud user patches and maintains the operating systems and the application software. Cloud providers typically bill IaaS services on a utility computing basis: cost reflects the amount of resources allocated and consumed.

Cloud clients access cloud computing using networked client devices, such as desktop
computers, laptops, tablets and smartphones an d any Ethernet enabled device such as Home Automation Gadgets. Some of these devices – cloud clients – rely on cloud computing for all

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

or a majority of their applications so as to be essentially useless without it. Examples are thin clients and the browser-based Chromebook. A Cloud Provider Interface (CPI) provides an abstraction from an underlying IaaS by defining a set of functions for managing virtual machines life-cycle in which might run an elastic service. When workload of an application increases while the provisioned resources are not reconfigured in time to satisfy the changed performance requirements, the performance of the application will degrade. The degradation of performance may degrade the Quality of Service (QoS) and the Service Level Objective (SLO) may be thus violated. This problem could be critical because one direct consequence of performance degradation is losing users.

ELASTIC resource provisioning is one of the most important and attractive features of modern IaaS cloud system. However, virtualized resource management is still a challenging task because

1) Optimal resource allocation has to be selected out of a large number of alternatives
2) The workloads of hosted services may vary enormously over time
3) The workload of some applications may not grow or decrease tremendously during a very short period whereas some do.

Existing approaches exploit automated resource manager/ scheduler to overcome these problems. Christo Ananth et al. [4] discussed about a method, Optimality results are presented for an end-to-end inference approach to correct(i.e., diagnose and repair) probabilistic network faults at minimum expected cost. One motivating application of using this end-to-end inference approach is an externally managed overlay network, where we cannot directly access and monitor nodes that are independently operated by different administrative domains, but instead we must infer failures via end to-end measurements. We show that first checking the node that is most likely faulty or has the least checking cost does not necessarily minimize the expected cost of correcting all faulty nodes. In view of this, we construct a potential function for identifying the candidate nodes, one of which should be first checked by an optimal strategy. Due to the difficulty of finding the best node from the set of candidate nodes, we propose several efficient heuristics that are suitable for correcting fault nodes in large-scale overlay networks. We show that the candidate node with the highest potential is actually the best node in at least 95% of time, and that checking first the candidate nodes can reduce the cost of correcting faulty nodes as compared to checking first the most likely faulty nodes.

The disadvantages here include

1) Although these approaches can adjust the resource allocations to meet the time-varying performance requirement, the QoS of the hosted applications still cannot be guaranteed especially when workload increases substantially in a short period
2) During the series of adjustments, however, the SLO may have been violated
3) Some approaches always require tuning offline or even manually and need re-tuning or recalibrating when workload characteristics change.

A novel resource management framework has been introduced here to ensure high-level QoS in the cloud computing system. This framework utilizes an aggressive resource provisioning strategy which encourages to substantially increase the resource allocation in each adaptation cycle when workload increases. This strategy first provisions resources which are possibly more than actual demands, and then reduces the over-provisioned resources if needed. By applying the aggressive strategy, the framework can satisfy the increasing performance requirement in the first place so that the QoS can be kept at a high level.Given the problems, a framework is proposed that dynamically adjusting the number of VM instances to ensure the QoS by accelerating the resource provisioning in virtualized cloud computing environments.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

II. RELATED WORK

A. Virtual machine provisioning based on analytical performance and qos in cloud computing environments

Cloud computing is the latest computing paradigm that delivers IT resources as services in which users are free from the burden of worrying about the low-level implementation or system administration details. At runtime, there may be unpredictable situations obstructing the smooth provisioning and delivery of application services such as

a. Estimation error
b. Highly dynamic workload
c. Uncertain behavior

To over the problem described above "an adaptive provisioning technique" is used here. This provisioning technique automatically adapts to workload changes related to applications for facilitating the adaptive management of system and offering end users guaranteed Quality of Services (QoS) in large, autonomous, and highly dynamic environments. The behavior and performance of applications and Cloud-based IT resources are modeled to adaptively serve end-user requests. To improve the efficiency of the system, analytical performance (queueing network system model) and workload information to supply intelligent input about system requirements to an application provisioner with limited information about the physical infrastructure are used.The contributions here are:

- An adaptive provisioning technique based on analytical performance and workload information for dynamically determining and capturing the relationship between application QoS targets and the allocation of individual IT resources. This technique captures the complex behaviour of applications including requests arrival rates and resource demands over time
- An analysis of two well-known application-specific workloads aimed at demonstrating the usefulness of workload modeling in providing feedback for Cloud provisioning
- A comprehensive simulation-driven analysis of the approach based on realistic and well-known production environment workload models

Here VM Provisioning and Application Provisioning are focused, because these are the steps that application service providers can control. The goal of Application Provisioning is ensuring an efficient utilization of virtualized IT resources, which can be achieved through the use of techniques such as load balancing and efficient mapping of requests, while the goal of VM Provisioning is to provide applications with sufficient computational power, memory, storage, and I/O performance to meet the level of QoS expected by end-users. The latter is achieved either by increasing/decreasing capacity of deployed virtual machines or by increasing/decreasing the number of application and VM instances.The issues here are (1) Modeling and decision-making processes used by the mechanism will be improved to support not only changes in number of VMs but also changes in each VM capacity (2) Intend to improve the queueing model to allow modeling composite services and access to Cloud storage (3) Need to adapt more comprehensive prediction techniques to handle prediction for arbitrary service workloads

B. DEJAVU: Accelerating resource allocation in virtualized environments

DejaVu is a framework proposed here that

a. Minimizes the resource management overhead by identifying a small set of workload classes for which it needs to evaluate resource allocation decisions
b. Quickly adapts to workload changes by classifying workloads using signatures and caching their preferred resource allocations at runtime
c. Deals with interference by estimating an "interference index".

This framework addresses this set of problems by proposing DejaVu, a system that simplifies and accelerates the management of virtualized resources in cloud computing services. The key idea behind DejaVu is to cache and reuse the results of previous

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

resource allocation decisions. When the DejaVu framework detects that workload conditions have changed (perhaps because a VM or service is not achieving its desired performance), it can lookup the DejaVu cache, each time using a VM identification and a workload signature. The signature is an automatically determined, pre-defined vector of metrics describing the workload characteristics, and the VM's current resource utilization. The contributions are as follows:

- Propose DejaVu, a framework for learning and reusing optimized VM resource allocations.
- Describe a technique for automatically profiling, clustering, and classifying workloads. Clustering reduces the number of tuning instances and thus reduces the overall resource management overhead.
- Evaluate DejaVu using realistic network services and real world MSN messenger and HotMail traces. Our results show that DejaVu achieves more than 10x speedup in adaptation time for each workload change, relative to the state-of-the-art.

The issue arises here is DejaVu only records the optimal resource allocations rather than finding them. If a workload has never been seen before, DejaVu only resorts to maximum resource allocation to maintain the performance. Hence, still needs traditional efforts to find optimal resource allocations.

C. A distributed self-learning approach for elastic provisioning of virtualized cloud resources

A reinforcement learning algorithm has been developed with a highly efficient representation of experiences as the heart of the VM side learning engine. The mechanism and the distributed learning algorithm have been prototyped in an iBalloon system. This technique addresses the issues and presents a distributed learning mechanism for cloud management. More specifically, our contributions are as follows:

- Distributed learning mechanism: VM resource allocation is treated as a distributed learning task. Instead of cloud resource providers, cloud users manage individual VM capacity and submit resource requests based on application demands. The host agent evaluates the aggregated requests on one machine and gives feedback to individual VMs. Based on the feedbacks, each VM learns its capacity management policy accordingly. The distributed approach is scalable because the complexity of the management is not affected by the number of VMs and we rely on implicit coordination between VMs belonging to the same virtual cluster.
- Self-adaptive capacity management: An efficient reinforcement learning approach for the management of individual VM capacity has been developed. The learning agent operates on a VM's running status which is defined on the utilization of multiple resources. There employ a Cerebellar Model Articulation Controller-based Q table for continuous state representation. The resulted RL approach is robust to workload changes because state on low-level statistics accommodate workload dynamics to a certain extent.
- Resource efficiency metric: Explicitly optimize resource efficiency by introducing a metric to measure a VM's capacity settings. The metric synthesizes application performance and resource utilization. When employed as feedbacks , it effectively punishes decisions that violate applications' SLA and gives users incentives to release unused resources.
- Design and implementation of iBalloon: The prototype implementation of the distributed learning mechanism, namely iBalloon, demonstrated its effectiveness in a Xen-based cloud testbed. iBalloon was able to find near optimal configurations for a total number of 128 VMs on a 16-node closely correlated cluster with no more than 5% of performance overhead . It would be noted that, there were reports in literature about the automatic configuration of

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

multiple VMs in a cluster of machines. This is the first work that scales the auto-configuration of VMs to a cluster of correlated nodes under work conserving mode.

The problems here are that the iBalloon mainly focuses on scaling up the system horizontally (by increasing the VM number) and it only resorts to the optimal allocations based on acquired knowledge, which might be suboptimal because of the defectiveness of the knowledge

III. System Model

The framework is designed to work along with the services deployed in the cloud. It accelerates the virtualized resources provisioning by

1) Continuously learning the workloads and their associated resource requirements
2) Making bold attempts to provision sufficient resources when the performance of service degrades.

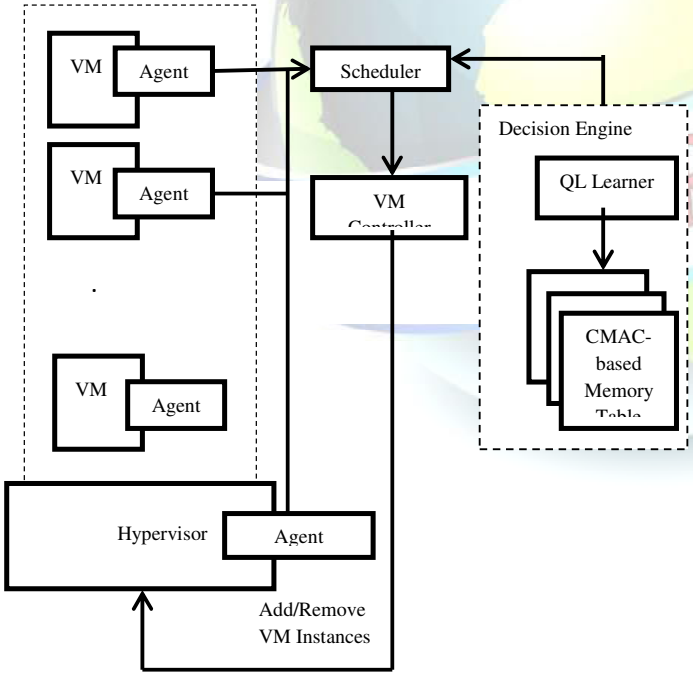

Fig. 1 System Architecture

The framework consists of three main components:

A. Agent

To collect load metrics and profile the workload, it employs a proxy that resides inside either the VM or the physical machine(PM). The proxy periodically collects the load metrics (e.g. CPU and memory utilization), and then sends them to the scheduler.

B. Scheduler

The scheduler calculates the Current Service Load Status (CSLS), which indicates the performance of a service during the previous time period, and then sends it to the Decision Engine to make a schedule decision. If there is more than one service hosted in the cloud, CSLS is calculated separately for each service.

C. Decision Engine

At the core of the Decision Engine is an RL-based learner called RLearner. It first generates a decision (an action to be taken to reconfigure the resource allocations) based on these metrics and existing knowledge, and then updates the knowledge with a reward that indicates the effectiveness this decision. The aforementioned knowledge, which is actually the mapping between workload metrics and actions, is stored in a table called Cerebellar Model Articulation Controller (CMAC) Table. The CMAC Table is a CMAC based memory table which will be updated every time an action is chosen and applied to the system. The CMAC Table is updated by an aggressive reward strategy to make the decision engine capable of adjusting the resource provisioning efficiently. With the utilization of the aggressive reward strategy, the actions that substantially increase resource provisioning are encouraged to be selected by the Decision Engine when the resources are not sufficiently provisioned.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

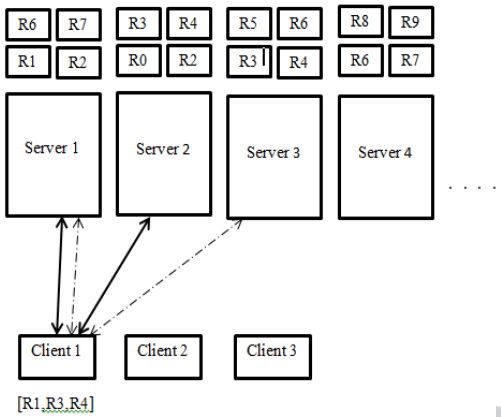

Fig. 2 An Audacious Resource Management Scenario

IV. SYSTEM IMPLEMENTATION

A. *Cloud Client*

Cloud collaboration allows clients and cloud applications to simultaneously use services from and route data among multiple clouds. This framework supports universal and dynamic collaboration in a multicloud system. It lets clients simultaneously use services from multiple clouds without prior business agreements among cloud providers, and without adopting common standards and specifications. The Cloud details should be maintained with the cloud virtual server. The server needs the information such as cloud IP Address, MAC Address etc., These details maintained with the authentication service. The cloud client needs to access the cloud virtual server with authorization details. With the authentication code provided by the agent through a secure transmission only the cloud client can access the service from the server.

B. *Cloud Virtual Server*

Infrastructure-as-a-Service like Amazon Web Services provides virtual server to start, stop, and access and configure their virtual servers and storage. In this module, the Virtual cloud server has been created with the Agent. The cloud needs to fetch the documents from the cloud it provide the cloud service environment. This can be implemented with the help of VM ware workstation because the virtual network systems may use. The Network controller and the cloud client should be collaborated with the isolated bridge that is the details need to be authorized with the cloud server via the agent. The agent is responsible for date and service management by providing authentication and authorization services.

C. *Service Provider(Agent)*

The cloud server needs a provider to furnish all the services that are offered to the cloud client. It provides the data management, service management etc to the client provided by the cloud server. The cloud service provider makes ranking about the cloud server to obtain the quality of service.

D. *Collecting Workload Metrics*

An agent that runs either in the VM that hosts the services or the PM that hosts the VMs to periodically collect the metrics and profile the workload. The agent collects the status of the VM and sends it to the scheduler for workload characterization. The status is represented by performance metrics which are chosen as the workload signature.

E. *RL-Based Learning Engine*

RL is a problem faced by an agent who must learn behavior through trial-and-error interactions within a dynamic environment. The RL model consists of (1) a discrete set of states *s*, (2) a discrete set of actions *a* and (3) a reinforcement signal (or so called *reward*). The goal of the agent is to find a policy _ mapping states to actions, which maximizes the long term reward. Formally, the RL problem can be modeled as a *Markov Decision Process* (MDP). The decision engine is implemented as an RL-based agent, which continuously takes actions to change the amount of provisioned resources according to the current performance status of the system.

F. *CMAC-Based Q-Table*

For implementing the Q-learning algorithm, a memory table for storing the learned state-action relationship is indispensable. First, the memory space occupancy of the flat table is high. Second, the convergence speed of flat table, which is critical for online management approaches, is low so that it has to run a considerably long time to approximate the optimal policy. To accelerate convergence and improve space utilization, we borrow the design of Credit Assigned CMAC (CA-CMAC) as the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

implementation of the Q-Table. CMAC is a type of neural network with the features of generalization during the learning/training phase and improved space utilization. In CMAC, input signal consists of a certain number (called input dimension) of state variables. Each state variable is represented by a sensor and the sensor discretizes each variable into a finite one-dimension vector according to desired accuracy. The actual content of CMAC is stored in a memory table, which is implemented as a hash table for better space utilization. When an input signal comes, the state variables are sent to sensors, and for each discretized state variable, more than one sensor units are activated. The activated sensor units are the key to the hash table and thus multiple memory cells in the hash table are activated as well. Each memory cell contains a weight value indicating the weight of the *(s,a)* pair and the arithmetic sum of these weight values is as the output of CMAC.

V. CONCLUSION

The problem of efficient resource management remains a challenging task in IaaS clouds, particularly when workload increases at a high speed. A framework has been presented here which is a generic reinforcement learning-based aggressive virtualized resource management system for IaaS clouds. For the implementation of this system, the following work has been implemented so far: (1) Implementation of cloud client which need to be authorized by providing authorization code; (2) Implementation of an agent who is responsible for resource and service management. This will allow /deny the request from the cloud client. This will generate an authorization code randomly and send it to the cloud client via a secure transmission if the cloud client is valid; (3) implementation of cloud virtual server which shows the browse/upload window. Both the cloud virtual server and the agent are placed in the VM. Thus the cloud client can access the cloud virtual server only via the agent with authentication and authorization. The adaptation process can be finished with limited adjustments and the increasing performance requirement can be satisfied in the first place. As a consequence, the QoS can be ensured at a high level, and the SLO violation rate can be reduced. As far the implementation of cloud client and server in presence of the agent have been done. The agent here is responsible for resource and service management. The agent authenticates the client by providing authentication code at runtime which is generated randomly. With this authentication code only the client can access the services provided by the cloud virtual server.

REFERENCES

[1] Guiyi Wei, et al. (2010), 'A game-theoretic method of fair resource allocation for cloud computing services', The Journal of Supercomputing. 54(2): 252-269.

[2] Jinzhao Liu, Yaoxue Zhang, Yuezhi Zhou, Di Zhang, and Hao Liu. (2015), 'Aggressive Resource Provisioning for Ensuring QoS in Virtualized Environments', IEEE transactions on cloud computing, vol. 3, no. 2,

[3] Nedeljko Vasi, Dejan Novakovi, Svetozar Miuin, Dejan Kosti, and Ricardo Bianchini. (2012), 'Dejavu: accelerating resource allocation in virtualized environments', ACM SIGARCH Computer Architecture News, vol. 40, no. 1, pp. 423-436. ACM.

[4] Christo Ananth, Mona, Kamali, Kausalya, Muthulakshmi, P.Arthy, "Efficient Cost Correction of Faulty Overlay nodes", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 1,Issue 1, August 2015,pp:26-28

[5] Rodrigo N. Calheiros, Rajiv Ranjan, and Rajkumar Buyya. (2011), 'Virtual machine provisioning based on analytical performance and QoS in cloud computing environments', ICPP 2011, pp. 295-304. IEEE.

[6] Zhang, Qi, et al. (2014), 'Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud', IEEE transactions on cloud computing. vol. 2, No. 1. pp. 14-28. IEEE.

[7] Auto Scaling, http://aws.amazon.com/autoscaling/.

[8] Right scale, http://www.rightscale.com/.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 15, March 2016

M.Nishanthi pursuing M.E Computer Science And Engineering degree in M.I.E.T Engineering college, Trichy in 2016 and received B.E Computer Science And Engineering degree from Saranathan college of Engineering, Trichy in 2014.Email: nishamageshz@gmail.com