CADRAM - Cooperative Agents Dynamic Resource Allocation and Monitoring in Cloud Computing

M.Abdullah[1], Dr. M. Mohamed Surputheen[2]

E-mail: abdulcs1963@gmail.com *E-mail:* msurfudeen@yahoo.com

[1]Research Scholar, Department of Computer Science, Jamal Mohamed College, Trichy, Tamil Nadu, India

[2]Associate Professor Department of Computer Science, Jamal Mohamed College, Trichy, Tamil Nadu, India

[Affiliated to Bharathidasan University]

Abstract

Cloud computing platform is a shared pool of resources and services with various kind of models delivered to the customers through the Internet. The methods include an on-demand dynamically-scalable form charged using a pay-per-use model. The main problem with this model is the allocation of resource in dynamic. In this paper, we have proposed a mechanism to optimize the resource provisioning task by reducing the job completion time while, minimizing the associated cost. We present the Cooperative Agents Dynamic Resource Allocation and Monitoring in Cloud Computing CADRAM system, which includes more than one agent in order to manage and observe resource provided by the service provider while considering the Clients' quality of service (QoS) requirements as defined in the service-level agreement (SLA). Moreover, CADRAM contains a new Virtual Machine (VM) selection algorithm called the Node Failure Discovery (NFD) algorithm. The performance of the CADRAM system is evaluated using the CloudSim tool. The results illustrated that CADRAM system increases resource utilization and decreases power consumption while avoiding SLA violations.

Keywords: *Cloud computing, cooperative-agent system, resource provisioning, SLA*

1. INTRODUCTION

Cloud computing is a type of parallel and distributed computing system consisting of a group of computers connected in a network that are dynamically provided one or more computing resources based on the Service Level Agreements (SLA) created through agreement between the cloud consumers and Cloud Service Providers(CSP). Cloud computing is an internet-based computing platform in which a number of remote servers are networked to distribute the data-processing tasks storage and an online access to computing services and resources. It depends on sharing of resources to achieve consistency and economies of scale, similar to a utility like the electricity grid over a network. Cloud resources are dynamically allocated on demand while shared by multiple users. The main supporting technology that enables resource sharing is virtualization. Virtualization software allows a physical computing device to be electronically separated into one or more "virtual" devices, each of which can be easily employed and managed compute tasks. Virtualization provides all the supports required to speed up IT operations, and reduces cost by increasing infrastructure utilization.

Cloud computing builds upon advances of virtualization and distributed computing to support inexpensive usage of computing resources, enhanced resource scalability and on-demand services. Virtual Machine (VM) technology has been employed for resource provisioning. Hence VMs are allocated to the user based on type of the job. Resource scheduling is an important function of every computing platform and so for cloud computing. The cloud scheduler is an application which has the responsibility of allocating resources to the jobs submitted to the cloud. It contains all the necessary VM images to run users' jobs. All the incoming jobs are queued up in scheduler. The scheduler is executed periodically. At each moment, the scheduler performs five tasks:

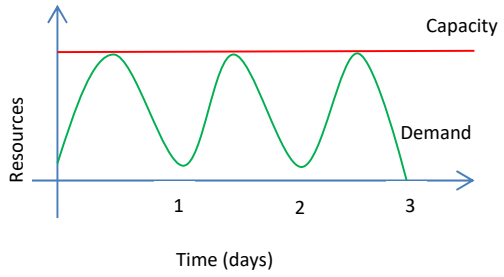
1) Predicting future incoming workloads;
2) Provisioning necessary VMs in advance, from clouds;
3) Allocating jobs to VM;
4) Releasing idle VMs if its Billing Time Unit (BTU) is close to increase;
5) If the time of un-allocated jobs is high, starting the necessary number of VMs.

1.1 Dynamic Resource Provisioning

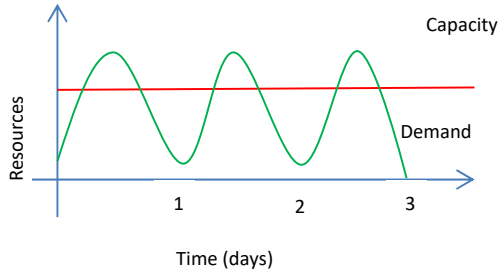
In cloud computing, *elasticity* denotes the level of automatic resource provisioning in response to the changes in the clients' demands. This is achieved by automatically, *scaling up* or *scaling down* the resources assigned to a particular client. This is a technology which automatically matches the clients' demand with the available resources in the CSP data center [1]. So, elasticity can simply support for avoiding the challenge in both the overprovisioning and the underprovisioning leading to good dynamic resource provisioning [2]. The *overprovisioning* problem occurs when the CDC allocates more resources for a particular client's job than the requested demands as shown in Figure 1 (a). On the when the allocated resources are less than the requested amount by a particular client then the

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.13

underprovisioning problem occurs. This situation is shown in figure 1(b). This problem leads to SLA violations and causes the loss of revenue and customers.



(a) Overprovisioning Problem



(b) Under-provisioning Problem

Fig.1. The overprovisioning and under provisioning problem in cloud

To address the resource provisioning problems as explained in the previous paragraph, dynamic resource allocating is employed [2]. The allocation of resources can be altered over time depending on the customer's demands as shown in Figure 2. This kind of on demand resource allocation strategies are already employed in several systems [3], [4]. One popular example is Amazon's Web Services (AWS) used on Amazon's EC2 cloud.

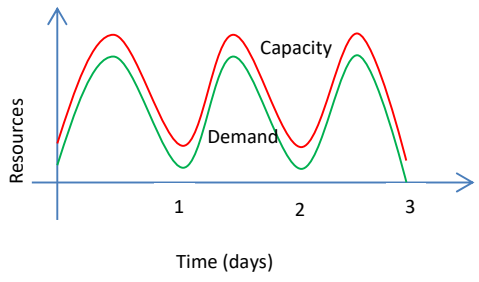


Fig.2. Dynamic resource allocation based on the client's demand

1.2 Cloud Resource Observation

Both resource allocation and observation in cloud computing are very important for both the CSPs and cloud clients. The cloud customers need to monitor their applications and the reserved Moreover, the resource monitoring includes gathering resource information and using this information to make decisions related to other components in the cloud environment [5].

The remainder of this paper has been organized as follows. Section 2 gives a brief review of related works regarding resource allocation in cloud environment. Section 3 presents the proposed algorithm for resource allocation and an overview of experimental environment. Section 4 shows the performance analysis of the proposed approach and finally Section 5 concludes the paper.

2. RELATED WORKS

There are several cloud computing organization that have exact technologies to observe and provide their resources. In this section, we will discuss some of the most popular frameworks in the field of resource allocation in cloud computing and various techniques used by them.

Venticinque et al. [6] have proposed the Open Cloud Computing Interface (OCCI) ramework which supports provisioning, observing and auto configuration of the cloud resources to satisfy the SLA at infrastructure level The OCCI includes a group of protocols and API which solve different problems in management of common tasks with satisfaction of integration, portability and interoperability requirements including autonomic scaling, deployment and monitoring.

Duong et al. [7], have proposed a framework called the CREATE framework architecture which is an extensible and reusable web service based framework with a group of resource provisioning algorithms for dynamic resource provisioning and adaptation on-demand under the IaaS cloud computing service model. There are three main components in CREATE which include resource sets (RS) monitoring and managements service (RMM), RS's adaptation service (RA) and Cloud clustering service (CC).

The Aneka system presented by Vecchiola et al. [8] is a .NET-based platform-as-a-Service (PaaS) application for cloud computing, which gives a set of APIs and runtime environment applications through multiple programming models, and implementing them on private and public cloud platforms like Amazon EC2. Siddiqui at al. [9], have proposed the Elastic-JADE system, which contains three parts like user, local machine and Amazon EC2 cloud. These three components allowed the system to automatically scale up or down. Amazon EC2's resources through JADE platform.

In [10] Naha et al have proposed a fog computing based resource allocation and provisioning algorithms by using resource ranking and provision of resources in a hybrid and hierarchical fashion. The proposed algorithms are evaluated in a simulation environment by extending the CloudSim toolkit to simulate a realistic Fog environment. Praveenchandar, et al [11] have proposed a dynamic resource allocation scheme based on the optimized task scheduling with improved power management

using prediction mechanism and dynamic resource table updating algorithm, efficiency of resource allocation in terms of task completion and response time is achieved.

Tang, et al [12] have proposed an edge based resource provisioning mechanism. The authors in this paper have considered the problem with respect to the resource location, the task priorities and the network transmission cost. In order to address this problem, the optimal problem is transformed to an optimal matching problem of the weighted bipartite graph.

Even though a number of researches have contributed towards dynamic resource allocation, it is still a challenge to recognize the load balancing for the computing nodes in the fog environment during the execution of IoT applications. In view of this challenge, Xu et al [13] have proposed a dynamic resource allocation method, named DRAM, for load balancing in fog environment.

3. PROPOSED SYSTEM

The CADRAM method has three main stages: observation, evaluation and implementation. The objective of observation stage is to observe the available resources in the CSP. The resources under observation are computational power or processor cycle. This can also be referred as CPU power, memory, storage, bandwidth of the network, etc. Apart from this observation stage also monitors the clients' jobs and requirements. The data collected from this stage are analyzed and evaluated in the second stage to enhance the resource administration by generating optimized decisions based on the VM's specification to be allocated for each client's request depending on the SLA. Finally, these decisions are forwarded to the third phase for implementation.

As shown in Figure 3, the CADRAM system contains two important layers: the client's applications layer and cloud service provider's (CSP) data center resources layer. The CSP resources include a set of datacenters each with a large number of physical machines called as hosts. Each host has different configurations with processors, memory, storage and network bandwidth to launch multiple VMs. Moreover, it has observing sensors to evaluate the resource consumption and the execution of customer's applications on it. The applications layer includes a set of customers having several jobs possibly employing multiple VMs. Each job is associated with a SLA which defines the task of every job such as time to complete their tasks, number of requests for specific time periods etc.

The CADRAM utilizes cooperative agents which includes a global agent and a set of local agents. Each client is assigned a local agent responsible for monitoring the clients' requests and determines the amount of resources based on a regression analysis of its history without violating SLA. The local agent forwards the formulated requests to the global agent, which is responsible for communicating with the hypervisors of the physical machines in order to allocate the actual resources in the different datacenters under the cloud provider's control.

The global agent should also consider the limitations on the provider's resources in addition to various performance metrics like energy utilization.

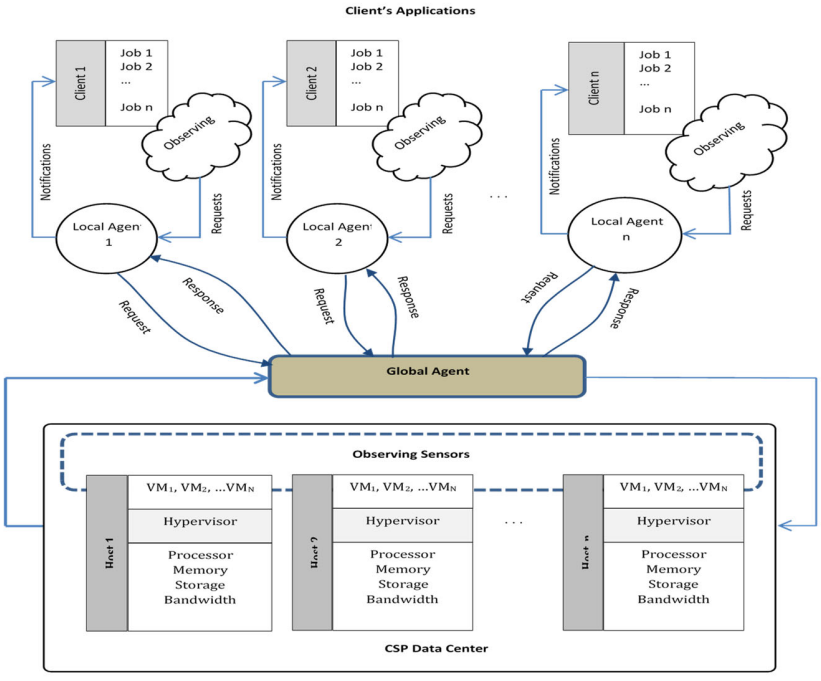


Fig.3 Architecture of CADRAM

3.1 The Local Agents

The key responsibility for every local agent is to observe the activities of the customer assigned to it and create a database of its resource utilization. Whenever a new request for a VM with specific characteristics is received from the customer, the local agent analyses the data base to restructure the client's request to match its actual usage. The aim of such restructure is to avoid both the over and under provisioning problems and also considering the other issues like the number of request per time unit, response time, etc. The restructured request is sent to the global agent which is responsible for the actual provisioning of the resources.

The local agents use simple regression analysis to restructure the clients' requests by forecasting the amount of wasted resources depending on the request. Here we explain how the actual required resources are calculated. For simplicity purpose we explain the process of calculating the CPU usage and the estimate for the other resources such as memory, storage and bandwidth can be calculated in the similar manner.

Eq.(1) shown below is used to determine the amount of the CPU that is actually be required by the client as a relation between the current and the previous requests made by the same client.

$$E_{CPU} = (R_{CPU} - W_{CPU}) + B \quad (1)$$

In this equation E_{CPU} represent the estimated CPU whereas R_{CPU} and W_{CPU} represent the currently requested and previously wasted CPU respectively. The variable B is used determine the amount of additional resources to be added based on the usage history of the client.

In this research, it is observed that overestimating the amounts of resources to be used causes wasting of resources, while underestimating them makes SLA violations. In general, most of the cloud users are ready to pay for wasting some resources but the jobs must be completed successfully and to avoid SLA violations. However, CSPs can have the control over the usage of resources and balance between the user requirements and allocation to keep up the SLA. The variable B will be assigned a high value if the CSP is more concentrating on SLA and customers are ready to pay for resources without bothering wastages. B will be assigned a small value if the CSP is caring about the resource utilization and don't want to waste the resources. Eq.(2) given below is used to calculated the wastage of CPU.

$$W_{CPU} = R_{CPU} X C_1 + C_0 \quad (2)$$

Where C_1 and C_2 are calculated using the Eq.(3) and (4) as given below:

$$C_1 = \frac{\sum_{i=1}^{c} [(Pi - \overline{P}) X (Wi - \overline{W})]}{\sum_{i=1}^{c} \left(Pi - \overline{P}\right) 2}$$

$$C_0 = \overline{W} - C_1 X \overline{P} \quad (4)$$

In the above equation P_i is the array containing the CPU requests made by the client.

3.2 The Global Agent

The global agent has the responsibility of observing the restructured requests coming from the local agents. This includes contacting the hypervisors in every physical server to provide the demanded resources, send the resource allocated report to the local agents and reporting the hypervisors about the completion of client's job. This will help the hypervisor to revoke the allocated resources. It is the responsibility of the global agent that it must consider and take into account the performance goals such as decreasing the number of physical machines and VM migrations and limiting the energy consumption of servers in CSP data center.

4. EXPERIMENTAL SETUP

We have used CloudSim simulator for implementing and testing our proposed method. We have used four types of VMs and two types of physical machines implemented in a CSP data center. The parameters of VMs and physical machines are given in Table I and II respectively.

Table I- Characteristics of VM

Number of VMs	50
Types of VMs	4
MIPS of CPU	2500,2000,1500,1000
Bandwidth	100 Mbs
Storage	2 GB

Table II- Characteristics of Physical Machine

Number of Hosts	50
Number Types	2
MIPS of CPU	1860, 2660
RAM	4 GB
Bandwidth	1 Gbits/S
Storage	100 GB

We have used the Node Failure Discovery (NFD) VM selection policy in this work. We have tested the NFD under several dynamic workload through several VMs allocation polices. We have compared the performance of NFD with other VM selection policies and the results are shown in the figure

There are a number of VM selection policies discussed in the earlier literature, However in this paper we have particularly considered the following four VM selection policies:

- The Maximum Correlation (MC) policy,
- The Minimum Migration Time (MMT) policy,
- The Minimum Utilization (MU) policy and
- The Random Selection (RS).

5. RESULTS AND DISCUSSIONS

We have considered several performance matrices for evaluating the selection policies such as VM migrations, energy consumption, host shutdowns and SLA violations.

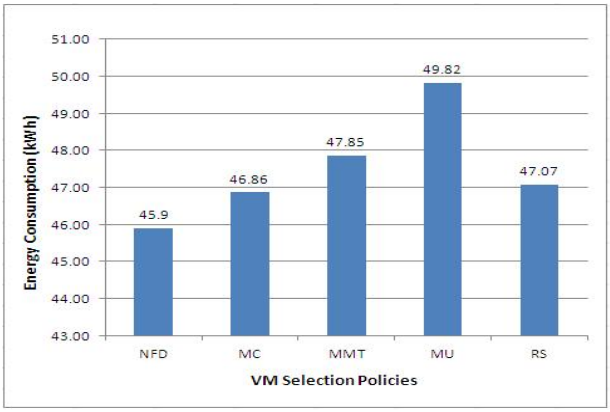


Fig.4. Comparison of the total energy consumption

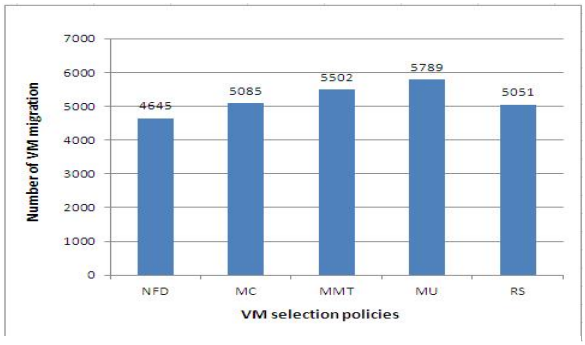


Fig.5. The number of VM migrations

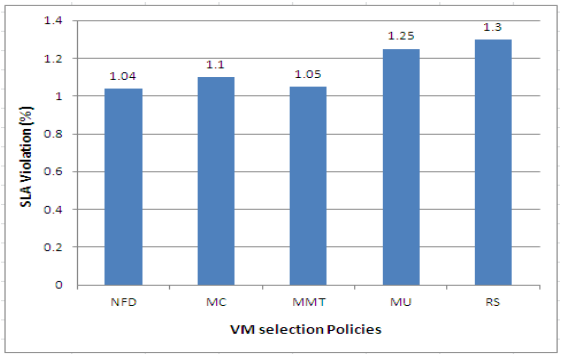


Fig.6. Percentage of SLA violations

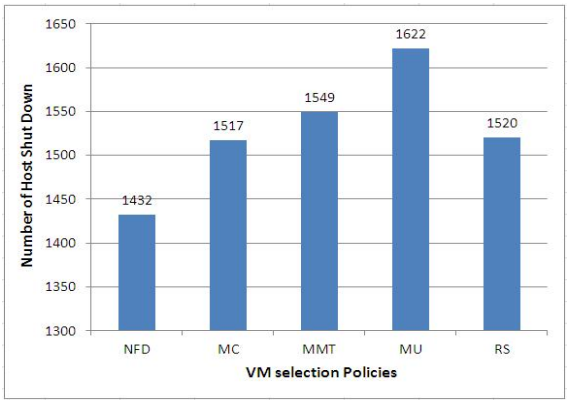


Fig.7. the number of host shutdowns

Figure 4 shows the comparison of the total energy consumption of the five VM selection policies. From these figure, the NFD is performing better when comparing with all other selection policies in terms of gain in power consumptions than other all policies. Figure 5 shows the number of VM migrations of the five selection policies and Similar to the energy consumption comparison, NFD outperforms all other selection policies with MC and RS being the closest to NFD. It is worth mentioning that there is a huge savings in terms of VM migrations. Figure 6 shows the percentage of SLA violations of the five VM selection policies. In this graph, we can observe that NFD is performing better than the all other selection policies in terms of SLA violation with respect to time and QoS. Figure 7 shows that the number of host shutdowns of the five selection policies under taken. As seen in the energy consumption comparison, NFD outperforms all other selection policies with respect to the number of host shutdowns.

6. CONCLUSION AND FUTURE RESEARCH DIRECTION

This paper presented the Cooperative Agents Dynamic Resource Allocation and Monitoring in Cloud Computing (CADRAM) system, to manage and observe the CSP's resources at the same time considering the client's quality of service (QoS) requirements defined in the service-level agreement (SLA). Moreover, CADRAM includes a new Virtual Machine (VM) selection algorithm called the Node Failure Discovery (NFD) algorithm. The proposed CADRAM system is evaluated using the CloudSim tool. The results demonstrate that the CADRAM system allows the cloud provider to improve the utilization of resource and decreases the energy consumption and avoiding SLA violations. In future we can extend this approach for fog computing based cloud computing extension with task offloading methods.

REFERENCES

[1] N. R. Herbst, S. Kounev, and R. Reussner, “Elasticity in cloud computing: What it is, and what it is not,” in *Proceedings of the 10thInternational Conference on Autonomic Computing (ICAC 2013), SanJose, CA*, 2013.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, “A view of cloud computing,” *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] P. Marshall, K. Keahey, and T. Freeman, “Elastic site: Using clouds to elastically extend site resources,” in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE Computer Society, 2010, pp. 43–52.

[4] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, “Dynamically scaling applications in the cloud,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 45–52, 2011.

[5] H. Huang and L. Wang, “P&p: A combined push-pull model for resource monitoring in cloud computing environment,” in *Cloud Computing(CLOUD), 2010 IEEE 3rd International Conference on*. IEEE, 2010, pp. 260–267.

[6] S. Venticinque, L. Tasquier, and B. Di Martino, “Agents based cloud computing interface for resource provisioning and management,” in *Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on*. IEEE, 2012, pp. 249–256.

[7] T. N. B. Duong, X. Li, and R. S. M. Goh, “A framework for dynamic resource provisioning and adaptation in iaas clouds,” in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 312–319.

[8] C. Vecchiola, X. Chu, and R. Buyya, “Aneka: a software platform for .net-based cloud computing,” High Speed and Large Scale Scientific Computing, pp. 267–295, 2009.

[9] U. Siddiqui, G. A. Tahir, A. U. Rehman, Z. Ali, R. U. Rasool, and P. Bloodsworth, “Elastic jade: Dynamically scalable multi agents using cloud resources,” in *Cloud and Green Computing (CGC), 2012 Second International Conference on*. IEEE, 2012, pp. 167–172.

[10] Naha, Ranesh Kumar, Saurabh Garg, Andrew Chan, and Sudheer Kumar Battula. "Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment." *Future Generation Computer Systems* 104 (2020): 131-141.

[11] Praveenchandar, J., and A. Tamilarasi. "Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing." *Journal of Ambient Intelligence and Humanized Computing* (2020): 1-13.

[12] Tang, Hengliang, Chunlin Li, Jingpan Bai, JianHang Tang, and Youlong Luo. "Dynamic resource allocation strategy for latency-critical and computation-intensive applications in cloud–edge environment.", *Computer Communications* 134 (2019): 70-82.

[13] Xu, Xiaolong, Shucun Fu, Qing Cai, Wei Tian, Wenjie Liu, Wanchun Dou, Xingming Sun, and Alex X. Liu. "Dynamic resource allocation for load balancing in fog environment." *Wireless Communications and Mobile Computing* 2018 (2018).