Volume 5, Issue 6, June 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Optimal Resource Provisioning Algorithm for Greener Future

G. Ezhilarasi[*]
Department of Computer Science
Pondicherry University
Karaikal Campus
Pondicherry, India

R. Brintha
Research Scholar
Department of Computer Science
Pondicherry University, kkl Campus
Pondicherry, India

S. Bhuvaneshwari
Head of the Department
Department of Computer Science
Pondicherry University, kkl Campus
Pondicherry, India

Abstract— Cloud computing is a distributed computing system that provide multiple users, wide range distributed access to scalable and virtualized environment over the internet. So far the researches has been made addressing how to provision the resources efficiently in the context of green computing. IAAS provides computing infrastructure and physical or virtual resources like network bandwidth, storage, and CPU. So IAAS is used to achieve better VM resources utilization efficiently. In this paper, Even though the optimum resources have been provisioned properly, the unutilized resources of the provided VMs are not green effective. Those unutilized resources would be the major reason for Energy waste. To minimizing both under provisioning and over provisioning problems under the demand and price uncertainty in cloud computing environments is our motivation to explore a resource provisioning strategy for cloud consumers. In particular, an optimal cloud Service Level Agreement (SLA) algorithm is proposed to minimize the total cost for provisioning resources in a certain time period. To make an optimal decision, the demand uncertainty from cloud consumer side and price uncertainty from cloud providers are taken into account to adjust the trade-off between on-demand and oversubscribed costs. The novel Cloud VM Creator enables automated VM Split to create Sliced VMs for the requested/user requirements. The approach allows IaaS users to always use fewest resources depending on the actual real-time workload. In comparison with using a fixed large-enough VM to manage fluctuated workload, it saves significant amount of computational resource, and enables better resource allocation and distribution inside IaaS Cloud. Thus the resources are utilizing more than resources is never waste and becomes greener future.

Keywords — Green computing, VM migration, IaaS, Splitter, SLA, CSP.

I. INTRODUCTION

In Cloud Computing, resources are provide to each user on demand and pay-per-use basis system and increase the opportunity for cloud users by accessing infrastructure and software application at any time. There are three cloud computing models, which are IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). IaaS provides computing infrastructure and physical or virtual resources like network bandwidth, storage, and CPU. PaaS provides computing platforms which typically include operating system, programming language execution environment, database, and web server. SaaS provides access to application software as on-demand software. All three types are delivered in four ways, which are publicly, privately, via a community, or in a hybrid cloud. Based on the models, service providers offer different types of services to cloud users with different demands. Cloud computing provides a pay-per-use payment. In other words, cloud users pay cost as much as using services. The major service providers, namely Amazon, Microsoft, and Google, offer many types of services and applications to cloud users through monitoring, managing, and provisioning resources. Amazon EC2 (Amazon Elastic Compute Cloud) and Microsoft Azure are examples of IaaS.

In IT environment the IaaS service manages the resource configurable computing resource is very small to dynamically provision and release minimal management effort. But, in cloud environment difficult to mention the different type of resources and reply changes in demands of users. Thus, the VM provisioning proper estimate and prepare the resource to meet dynamic demands of users.

VM provisioning is a strategy for managing resources by allocating resources on an "as needed" basis. It automatically adapts to workload changes related to applications for facilitating the adaptive management of system and offering end -users guaranteed QoS (Quality of Services). Typically, the provisioning is achieved by two operations - static and dynamic resource provisioning. In static resource provisioning, VMs are created with specified size and then consolidated onto a set of physical servers. The VM capacity does not change. In dynamic resource provisioning, VM capacity is dynamically adjusted to match workload fluctuations. Static provisioning often applies to the initial stage of capacity planning. In both static and dynamic provisioning, the estimation of the resource amount is one of the most important steps. The objective of the estimation is to ensure that VM capacity is commensurate with the workload. While over-provisioning wastes costly resources, under-provisioning degrades application performance and may violate SLA (Service Level Agreement).

Resource Provisioning Types

Based on the application needs they are classified as

1) **Static Provisioning**: For applications that have predictable and generally unchanging demands/workloads, it is possible to use “static provisioning" effectively. With advance provisioning, the customer contracts with the provider for services and the provider prepares the appropriate resources in advance of start of service. The customer is charged a flat fee or is billed on a monthly basis.

2) **Dynamic Provisioning:** In cases where demand by applications may change or vary, “dynamic provisioning" techniques have been suggested whereby VMs may be migrated on-the-fly to new compute nodes within the cloud. With dynamic provisioning, the provider allocates more resources as they are needed and removes them when they are not. The customer is billed on a pay-per-use basis. When dynamic provisioning is used to create a hybrid cloud, it is sometimes referred to as cloud bursting.

3) **User Self-provisioning:** With user self-provisioning (also known as cloud self-service), the customer purchases resources from the cloud provider through a web form, creating a customer account and paying for resources with a credit card. The provider's resources are available for customer use within hours, if not minutes.

II. RELATED WORK

In cloud computing, a resource provisioning mechanism is required to supply cloud consumers a set of computing resources for processing the jobs and storing the data. Cloud providers can offer cloud consumers two resource provisioning plans, namely short-term on-demand and long-term reservation plans. Cloud providers which offer IaaS services for both plans. In general, pricing in on-demand plan is charged by pay-per-use basis. Therefore, purchasing this on-demand plan, the consumers can dynamically provision resources at the moment when the resources are needed to fit the fluctuated and unpredictable demands. For reservation plan, pricing is charged by a onetime fee typically before the computing resource will be utilized by cloud consumer. With the reservation plan, the price to utilize resources is cheaper than that of the on-demand plan. In this way, the consumer can reduce the cost of computing resource provisioning by using the reservation plan.

Disadvantages- The reservation plan, the cloud consumers reserve the resources in advance. As a result, the under provisioning problem can occur when the reserved resources are unable to fully meet the demand due to its uncertainty. Although this problem can be solved by provisioning more resources with on-demand plan to fit the extra demand, the high cost will be incurred due to more expensive price of resource provisioning with on-demand plan. On the other hand, the over provisioning problem can occur if the reserved resources are more than the actual demand in which part of a resource pool will be underutilized. It is important for the cloud consumer to minimize the total cost of resource provisioning by reducing the on-demand cost and oversubscribed cost of under provisioning and over provisioning. To achieve this goal, the optimal computing resource management is the critical issue.

III. PROBLEM STATEMENT

In this paper, even though the optimum resources have been provisioned properly, the unutilized resources of the provided VMs are not green effective. Those unutilized resources would be the major reason for Energy waste. To minimizing both under provisioning and over provisioning problems under the demand and price uncertainty in cloud computing environments is our motivation to explore a resource provisioning strategy for cloud consumers. In particular, an optimal cloud Service Level Agreement (SLA) algorithm is proposed to minimize the total cost for provisioning resources in a certain time period. To make an optimal decision, the demand uncertainty from cloud consumer side and price uncertainty from cloud providers are taken into account to adjust the trade-off between on-demand and oversubscribed costs.

IV. PROPOSED SYSTEM

The novel **Cloud VM Creator** enables automated VM Split to create Sliced VMs for the requested/user requirements. This approach allows IaaS users to always use fewest resources depending on the actual real-time workload. In comparison with using a fixed large-enough VM to manage fluctuated workload, it saves significant amount of computational resource, and enables better resource allocation and distribution inside IaaS Cloud.

Through SLA, provider can share our unused VM to any other users. So, user can use unused VM’s to use other applications, and the results show that SLA can minimize the total cost under uncertainty. In this paper, we will discuss how cloud consumer can successfully minimize total cost of resource provisioning in cloud computing environments through SLA.

V. SYSTEM ARCHITECTURE

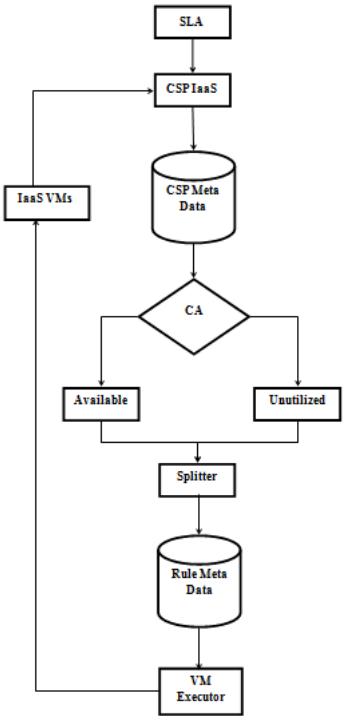


Figure 1: Architecture of Optimal Resource Provisioning

SLA-based Cloud Computing

A **service level agreement** (**SLA**) is interface between a service provider (either internal or external) and the end user that offers quality, responsibility and scope. Also provide negotiate document that define warranty for cloud (up time and down time).

The resource provisioning will be driven by market-oriented principles for efficient resource allocation depending on user following QoS targets and workload demand patterns:-

- Support for customer-driven service management based on customer profiles and QoS requirements.
- Definitions of computational risk management tactics to identify, assess, and manage risks involved in the execution of applications.
- Derivation of appropriate market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain SLA-based resource allocation.
- Incorporation of autonomic resource management models.
- Leverage of Virtual Machine technology to dynamically assign resource shares.
- Implementation of the developed resource management strategies and models into a real computing server.

CSP IaaS and CSP Meta data

Cloud Service provider (CSP IaaS) get the user requirements from SLA. Whatever the resources it store in CSP Meta data. And CSP Meta data is a database storage technique for which all the data's are stored and passes through according the condition.

Conditional Analyzer

The condition analyzer is decision making statement.it is working in two cases: one is Space is available or not and another one is Unutilized spaces. Then conditional Analyzer check for common cases the space is available are not if it is available then it gives to splitter otherwise it checks the unutilized space and gives to splitter. So that resources is provided more efficiently.

Splitter

Splitter Split the VM into 4 pieces according to the user requirements. According to that the rules are created and stored in rule Meta data. The Rule Meta Data is a repository of database. It maintains the rule created during migration and modification of VMs. Contains the automate VM translation when scaled VM always run in green boundary.

VM executor

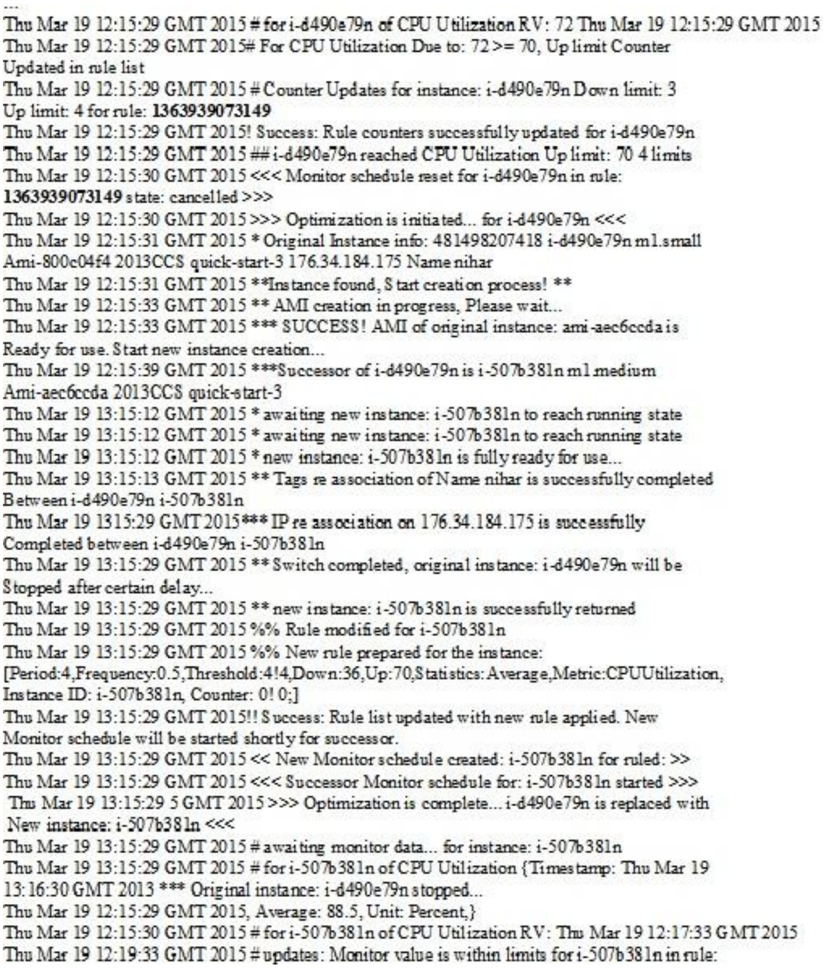
After splitter splits the VM and stored into the rule Meta data. And VM executor executed and created a new VM is called IaaS VM. The created VM gives to CSP IaaS. From CSP it store in CSP Meta data.

Algorithm for Optimal Resource Provisioning:

```
Input get from user
Initialize scheduling, UL, DL, period, frequency, use, VM size, x;
Initialize all the VM allocation status to AVAILABLE in the VM state list;
```

```
If (CSP entry of a VM corresponding to the current requesting user base)
        Then
The VM is allocated to the user base request;
Else if
Any unutilized space splitter split according to user based request
 Check VM size >= input
 Get VM
                X = VM size/ 4;
                 For each VM utilization
                {
                        If CPU Utilization > UL
                        {
                        DL updated;
                        Use = UL;
                        }
                End if
                Else if   CPU utilization<DL
                        {
                        UL updated;
                        Use = DL;
                        }
                End else if
                Updated in Rule Meta data;
                End for;
Available = VM total –use;
Else
Allocated the available space;
Update the entry of the user base and the VM in the CSP and the VM state list;
}
```

VI. IMPLEMENTATION



```
...
Thu Mar 19 12:15:29 GMT 2015 # for i-d490e79n of CPU Utilization RV: 72 Thu Mar 19 12:15:29 GMT 2015
Thu Mar 19 12:15:29 GMT 2015# For CPU Utilization Due to: 72 >= 70, Up limit Counter
Updated in rule list
Thu Mar 19 12:15:29 GMT 2015 # Counter Updates for instance: i-d490e79n Down limit: 3
Up limit: 4 for rule: 1363939073149
Thu Mar 19 12:15:29 GMT 2015! Success: Rule counters successfully updated for i-d490e79n
Thu Mar 19 12:15:29 GMT 2015 ## i-d490e79n reached CPU Utilization Up limit: 70 4 limits
Thu Mar 19 12:15:30 GMT 2015 <<< Monitor schedule reset for i-d490e79n in rule:
1363939073149 state: cancelled >>>
Thu Mar 19 12:15:30 GMT 2015 >>> Optimization is initiated... for i-d490e79n <<<
Thu Mar 19 12:15:31 GMT 2015 * Original Instance info: 481498207418 i-d490e79n m1.small
Ami-800c04f4 2013CCS quick-start-3 176.34.184.175 Name nihar
Thu Mar 19 12:15:31 GMT 2015 **Instance found, Start creation process! **
Thu Mar 19 12:15:33 GMT 2015 ** AMI creation in progress, Please wait...
Thu Mar 19 12:15:33 GMT 2015 *** SUCCESS! AMI of original instance: ami-aec6ccda is
Ready for use. Start new instance creation...
Thu Mar 19 12:15:39 GMT 2015 ***Successor of i-d490e79n is i-507b381n m1.medium
Ami-aec6ccda 2013CCS quick-start-3
Thu Mar 19 13:15:12 GMT 2015 * awaiting new instance: i-507b381n to reach running state
Thu Mar 19 13:15:12 GMT 2015 * awaiting new instance: i-507b381n to reach running state
Thu Mar 19 13:15:12 GMT 2015 * new instance: i-507b381n is fully ready for use...
Thu Mar 19 13:15:13 GMT 2015 ** Tags re association of Name nihar is successfully completed
Between i-d490e79n i-507b381n
Thu Mar 19 1315:29 GMT 2015*** IP re association on 176.34.184.175 is successfully
Completed between i-d490e79n i-507b381n
Thu Mar 19 13:15:29 GMT 2015 ** Switch completed, original instance: i-d490e79n will be
Stopped after certain delay...
Thu Mar 19 13:15:29 GMT 2015 ** new instance: i-507b381n is successfully returned
Thu Mar 19 13:15:29 GMT 2015 %% Rule modified for i-507b381n
Thu Mar 19 13:15:29 GMT 2015 %% New rule prepared for the instance:
[Period:4,Frequency:0.5,Threshold:4!4,Down:36,Up:70,Statistics:Average,Metric:CPUUtilization,
Instance ID: i-507b381n, Counter: 0! 0;]
Thu Mar 19 13:15:29 GMT 2015!! Success: Rule list updated with new rule applied. New
Monitor schedule will be started shortly for successor.
Thu Mar 19 13:15:29 GMT 2015 << New Monitor schedule created: i-507b381n for ruled: >>
Thu Mar 19 13:15:29 GMT 2015 <<< Successor Monitor schedule for: i-507b381n started >>>
 Thu Mar 19 13:15:29 5 GMT 2015 >>> Optimization is complete... i-d490e79n is replaced with
 New instance: i-507b381n <<<
Thu Mar 19 13:15:29 GMT 2015 # awaiting monitor data... for instance: i-507b381n
Thu Mar 19 13:15:29 GMT 2015 # for i-507b381n of CPU Utilization {Timestamp: Thu Mar 19
13:16:30 GMT 2013 *** Original instance: i-d490e79n stopped...
Thu Mar 19 12:15:29 GMT 2015, Average: 88.5, Unit: Percent,}
Thu Mar 19 12:15:30 GMT 2015 # for i-507b381n of CPU Utilization RV: Thu Mar 19 12:17:33 GMT 2015
Thu Mar 19 12:19:33 GMT 2015 # updates: Monitor value is within limits for i-507b381n in rule:
```

Figure 2: System Logs

This paper provides optimized the VM and splitter split the size of the VM according to the user requirements and rule. After rule is created then alter /delete is made by clicking the start/restart windows. Each VM having own rule that run in the green boundary. When over the optimization occurs, each user details (IP address, name) is transferred to successor. So that because of splitter resources should be provision efficiently. This below figure demonstrated in an example of logged saved when optimization is occurs.

VII. EVALUATION

First fig shows the workload pattern .fig2 shows memory utilization of the job. It configuration of memory is 32 GB.

Fig3 shows that Memory utilization of the job using optimal resource algorithm. Initially the Qatar amount of requested resource provision to the user and given the physical axes.by monitoring the usage of a user the threshold limit checks periodically. When it reach the threshold limit the y consequently the slice will be provided automatically and the rule has been updated. Likewise the process goes on. Here the job A requested 16 GB of memory where he has been provided 4 GB initially. And the threshold limit set as 0.8! 32, this job consumes 3.2GB memory at times 4.5. Therefore the second slice 8 GB of memory has been provided to the job A.and the threshold limit automatically updates 1.6!6.4.

By this evaluation it proved the memory utilize efficiently by providing the resources according to the usage of job rather than request user made.therfore unutilized memory can be provided to the new requested at the peak time.

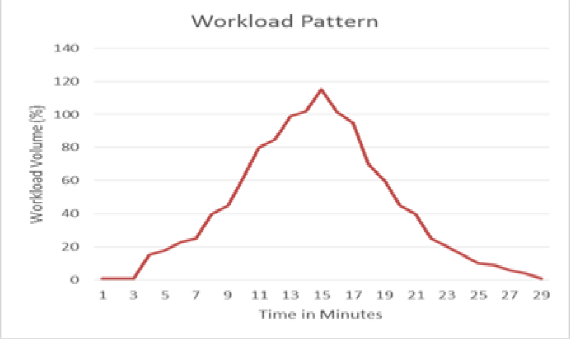


Figure 3: Workload Pattern

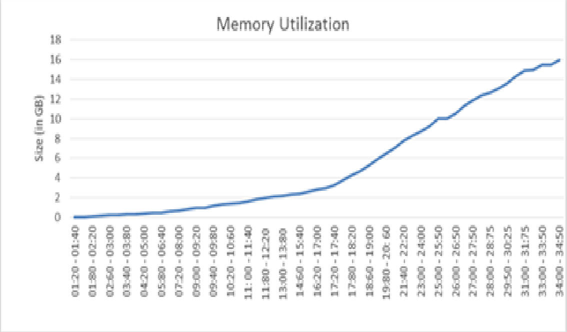


Figure 4: Memory Utilization

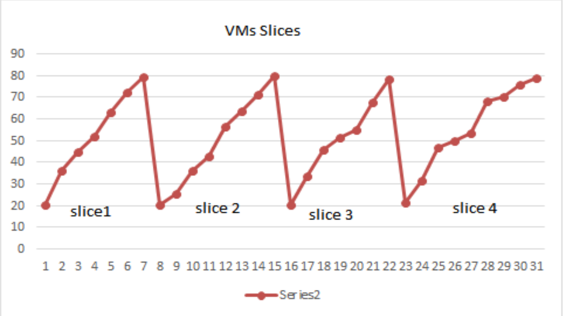


Figure 5: VM Slicing- Percentage of Utilization – Adding up the VM Slices

VIII. CONCLUSION AND FUTURE WORK

In Cloud Computing, Resource Provisioning means the selection, deployment, and run-time management of software (e.g., database servers, load balancers etc.) and hardware resources (e.g., CPU, storage, network etc.) for ensuring guaranteed performance for applications. These techniques are used to improve response time, performance, save energy, QoS, SLA. The ultimate goal of Resource Provisioning is to maximize profit from the Cloud Service Provider's Perspective and from the Cloud User's Perspective to reduce cost.

There are many challenges in the existing Resource Provisioning strategies. A mechanism that overcomes the challenges of the existing techniques has to be used. Mechanisms have to be proposed to efficiently make of cloud resources so that QoS is met and SLA violation in minimized in clouds when dynamically provisioned. Also these provisioning mechanisms must be used for IaaS users.

REFERENCES

[1] Amazon.com, "Amazon Web Services (AWS)," Online at http://aws. amazon.com, 2008.

[2] www.wikipaedia.com

[3] Torry Harris, "Cloud Computing – an overview".

[4] Rajkumar Buyya, Saurabh Kumar Garg, and Rodrigo N. Calheiros, "SLA – Oriented Resource Provisioning for cloud Computing: Challenges, Architecture and Solutions, paper 978-1-4577-1637-9/11, 2011, IEEE

[5] Bhavani B H and H S Guruprasad, "Resource Provisioning Techniques in Cloud Computign Environment", Vol 3, Issue 3, 2014.

[6] J. Udayakumar, M.Manikkam, A.Arun, "Cloud – SLA: Service Level Agreement for Cloud Computing", Vol.1 (02) 13-16, ISSN: 2278-9669, 2013.

[7] Rohit Vijayan, Bijolin Edwin, "Survey on Existing Resource Provisioning Techniques available in Cloud for effective utilization", Vol. 4, Issue 1, 2014.