



Intelligent & Automated VM Green Optimization

¹Nihar Ranjan Nayak*, ²R. Brintha, ³S. Bhuvaneshwari

²Research Scholar, ³Head of the Department,

^{1, 2, 3}Department of Computer Science, Pondicherry University,
Karaikal Campus, Pondicherry, India

Abstract — Cloud computing is a network of server's technology that relies on-demand sharing computing resources rather than the personal device to handle application of the internet on pay-per-use basic system and provide reliable, customized and QoS (Quality of Service). Previously transitions occurs step-by-step process so that optimization is affected and Much GCC research has been done addressing on virtualization and VM management optimizations. The users used to request for the additional resources so as to keep them in guaranteed safe mode. At this point, when the requested resources has been provisioned exactly, the unutilized resources kept idle. Those unutilized resources would be the major reason for Energy waste. The novel VM optimization based on the user's utilization enables automated VM size migrated where the scaled VMs always run within their own specified green boundaries no matter how individual workload varies. The IAVGO process act of finding appropriate VM successor using auto scaling to allocates the VM in most green effective size. Optimal resources provisioning for greener future using automated rule, IAVGO enable to provide better CPU utilization and it paves the way for the requesting users to get the resources even in the peak time. So resources never be wasted then environment becomes greener.

Keywords — Green Optimization, CSP, VM Executor, IAAS VM, Rule Metadata, load-balancing algorithm.

I. INTRODUCTION

In cloud computing systems virtual machines (VMs) provide flexibility and portability through easy allocation, which enables continuous mapping of VMs to available resources. These approach can only be implemented in CSP Metadata (Cloud Service provider) and the Metadata is using in data about data or repository of data. It working just like database storage. IAVGO is developed to work with IaaS VM and Rule Metadata by allowing green boundary using automatic optimization. These automated optimization technique provides multiple resources provisioning plan as well as long term plan using for improving performance. As Intelligence and automated based solutions, IAVGO monitor the CPU utilization periodically of the VM used by user and through checker it updated the VM. The whole process defined intelligent optimization rule and the proposed approach contains:

- Through SLA based agreement is negotiate document using up-time, down-time the analyzer split resource according to user requirement and working automated using Rule Metadata.
- For improving resources utilization efficiently in the process of greener environment, virtualization provides various types of computational resources. In fact infrastructure-as-a-service (IaaS) users can achieve the VM resource utilization efficiently in better way.
- The process of greener system, we reduce resource consumption and disposal of electronic waste. This is the very important feature of green optimization technique.
- The repository Rule Metadata is using optimization technique migrate resource those is in green boundary (peak time).and update the resources.

II. RELATED WORK

The author propose the optimize VM scheduling techniques over cloud data centre based on resource provisioning and allocation algorithm, where green cloud creature involving "energy distinction technique and power model" is demonstrated. Earlier work based on similar approaches can also be founded. Nonetheless, they all share limitation:

1. Time and cost management system of VM migration or state changes
2. Alternate/ Modify of workload pattern there should be overall effectiveness of the optimization affected.
3. While upsizing/ downsizing the original download up limit may become too large/small once certain optimization occurs and step wise infinite transition occur. So overall performance affected.

Dynamic resource allocation based on VM migration attempts to minimize the overall number of running PMs in a server very small according to relevant load pattern. Here, according to the splitter and rule metadata it allocates the VM size on demand users. Splitter split VM size according to user requirement through SLA based agreement. 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. So created IaaS VM is after the allocation migration is made, so different load VM migrated between the Physical Machine. Where the idle PM is turn off. After all

completed those in green boundary allocated the VM size. So resources are utilizing efficiently. For hardware/software combined optimizations, VM scheduling tasks to achieve better resource utilization efficiently and it focuses on improving resource utilization and consumption of VM provision in the cloud.

SYSTEM ARCHITECTURE

IaaS VMs

It is also called as created VM or splitter VM. Because splitter create the VM whatever the user needed. When the IaaS VM is created and gives to rule metadata and create the rule. For the created VM splitter migrates the VM size. After that created **IaaS VM**, According to rule it goes to the utilization monitor for verifying threshold value.

Utilization Monitor

Utilization Monitor calculates the speed of CPU utilization and monitors the VM which is given to the user. In two cases it checks: 1. Check after all four pieces are used by the user. 2. In peak time, using utilization monitor we can check it is green efficiency or not. It also checks threshold-up and threshold-down. For inefficient resources utilization IAVGO behaves perfectly to implement intelligent VM size allocation, according to optimization rule.

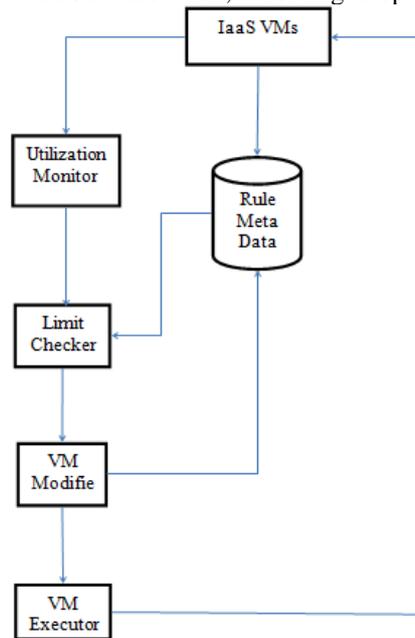


Fig 1: (Intelligent & Automated VM Green Optimization)

Limit Checker

It is the interface between utilization monitor and rule metadata. According to the monitor value or rule, the Limit Checker check whether the VM crossing the threshold value or not. Whenever the limit has been crossed VM size is migrated. Limit checker is automated rule based system and following the rule it goes through the VM modifier, executor and get modify by the scaling.

VM modifier and VM executor

VM is modified according to the rule and passes to the VM executor. And VM is executed the IaaS VM, store in rule Meta data. Also it is automatically update the rule. VM modifier contains the complete specification of VM and current VM images.

Rule Meta Data

The Rule Meta Data is a repository of database.it maintain the rule created during migration and modification of VMs. Contains the automate VM translation when scaled VM always run in green boundary. According to the CPU utilization the rule Meta data migrate the VM those who in the peak time that's all happens followed by rule.

III. OPTIMIZATION RULE

RULE PARAMETERS

The optimization rule has two parts. The first section is when rule is created (name, id etc.) and second is the body section that contains: Period, Frequency, Threshold up, Threshold down, Statistics, Metric, Counter etc. "Period" express the likely repeating period of how often workloads increase/decrease at the concrete level. "Frequency" is using for extraction of workload. "Threshold up" and "Threshold down" values when pieces on that time resources are utilized more, down limit vary. And pieces off when resources are utilized less, up limit varies. "Statistics" contains the value of CPU Matric sampled. For example: Network data, disk read/write and CPU utilization etc. "counter" refresh every time when instance of time has gone over below/over its green limits.

CUL- current up limit, PIV- previous initial value CDL-current down limit

$$\text{new up limit} = \begin{cases} \text{CUL} * 70\% (\text{CIV} \leq 2 * \text{PIV}) \\ \text{CUL} * 90\% (\text{PIV} \geq \text{CIV}) \end{cases}$$

$$\text{new down limit} = \begin{cases} \text{CDL} * 70\% (\text{CIV} \geq 2 * \text{PIV}) \\ \text{CDL} * 90\% (\text{PIV} \leq \text{CIV}) \end{cases}$$

Thus the above equations illustrate that how the limit should be customized based on the current instant size and relevant previous initial value. For upsizing the up limit remain unchanged, where the current initial value less than double of previous initial value it gives current upper limit and vice-versa. In this way no matter how the workload scales up or down, it guaranteed that after the optimizing there is updated green boundary whether the workload will fall within the limit.

Optimization rule

The optimization rule has been specified with the metrics and name tags so as to maintain and update as per the usage and utilization of the resources by the jobs requested by the user. The sample rule tag is given to specify the IaaS optimization by the proposed intelligent system known as IAVGO. In this proposed system, The Rule Meta Data is a repository of database.it maintain the rule created during migration and modification of VMs. Contains the automate VM translation when scaled VM always run in green boundary. According to the CPU utilization the rule Meta data migrate the VM those who in the peak time that's all happens followed by rule.

```
<? xml version="1.0" encoding="UTF-8">
<rule type="Iaasoptimization">
<Provider>CSP </Provider>
<Rule id>1363939073149</Rule id>
<Name>nihar</Name>
<Time> Fri Mar 20 01:15:29 GMT 2015 </Time>
<vm id>i-d490e79n </vm id>
<optimization metric="CPUutilization">
<Period>4</Period>
<Frequency>0.5</Frequency>
<Up threshold>4</Up threshold>
<Down threshold>4</Down threshold>
<Up limit>70</Up limit>
<Down limit>36</Down limit>
</optimization>
</rule>
```

IV. ALGORITHM

Algorithm IAVGO ()

```
{
  Initialized CSP;
  CSP Allocates VM space into the VM state list;
  Do {
    While (VM space is available in VM state list)
    {
      Initialize all the VM allocation status;
      IF (new request are received by the CSP Meta data && current VM allocation status! =available)
      {
        CSP Meta data queue the requests;
        CSP Meta data removes a request from the beginning of the queue;
      }
      If (CSP contain any entry of a VM corresponding to the current requesting user base && VM allocation status ==
      AVAILABLE)
      {
        The VM is reallocated to the user base request;
      }
    }
    Allocate a VM to the user base request using Round Robin Algorithm;
    Update the entry of the user base and the VM in the CSP and the VM state list;
  }
  While (true);
}
```

V. IMPLEMENTATION

This paper provides a full optimization technique as well as IAAS interface. In the optimization panel one can create a novel optimization rule by using different types of parameter. After rule is created then alter/delete is made by clicking the start/restart windows and all the rule store in rule Meta data. Each VM having own optimization rule that run in automated optimization life cycle. The whole system run fully automatically and does not need any supervision. When over the optimization occurs, each user details (IP address, name) is transferred to successor. This is cleared that there no effect in optimization.

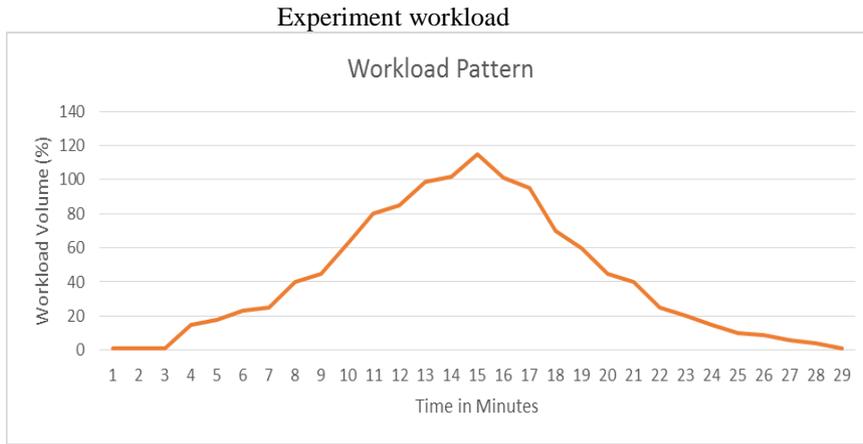
```
Thu Mar 19 12:15:29 GMT 2015 # for i-d490e79n of CPU Utilization RV: 72 Thu Mar 19 12:15:29 GMT 2015
Thu Mar 19 12:15:29 GMT 2015# For CPU Utilization Due to: 72 >= 70, Up limit Counter
Updated in rule list
Thu Mar 19 12:15:29 GMT 2015 # Counter Updates for instance: i-d490e79n Down limit: 3
Up limit: 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 13:15:29 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 rule: >>
Thu Mar 19 13:15:29 GMT 2015 <<< Successor Monitor schedule for: i-507b381n started >>>
Thu Mar 19 13:15:29 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 of IAVGO)

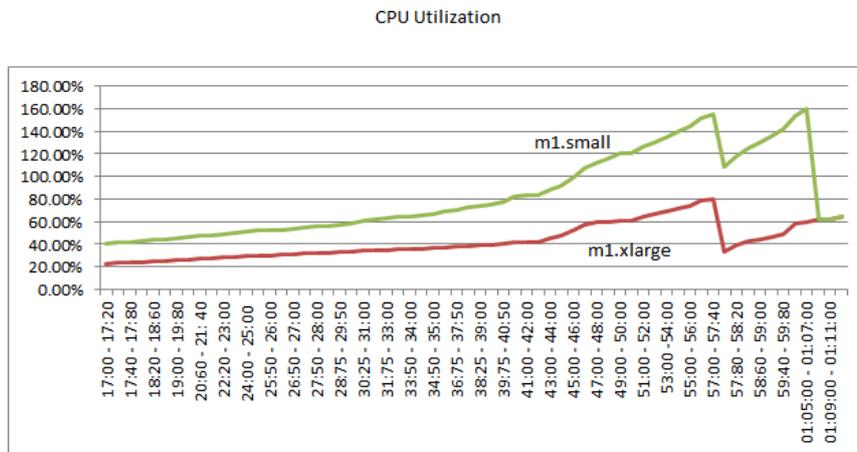
Thus the above figure demonstrated in an example of logged saved when optimization is occurs. As a matter of facts, the life-cycles of VM in clouds are different private infrastructure to environment. The time needed for VM creation, booting, image-creation, is highly valuable. This paper provides also lot of methods for error handling intelligently, using automated rule. Here AMI creation request is used for sent. If AMI is available the creation of successor is stored. When successor is fully available transformation is made involving the tag and IP association.

VI. EVALUATION

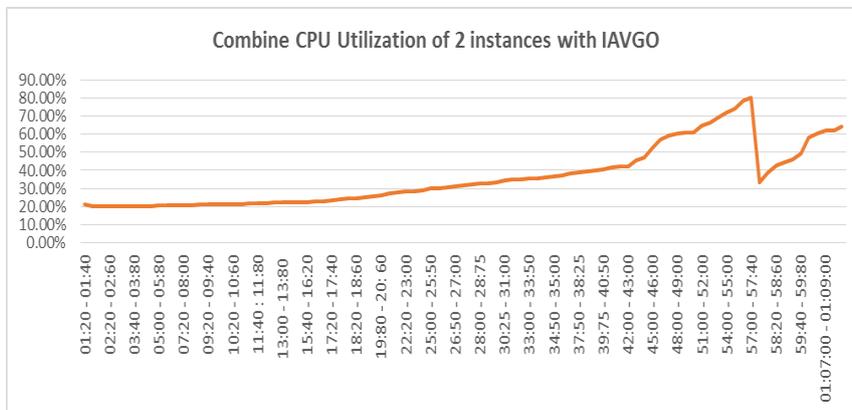
A series of experiments have been conducted in different VM size with different dynamics of distinct patterns. The test workloads were generated by Apache Jmeter that install on the clients machine, which sent quantitative server requests in a controlled manner. The workload pattern has been extracted and checked with the threshold limits. In server side, for every request Jmeter sent and calculation task executed. Each on utilized space, the IAAS provides both CPU core and process speed. so we adjust calculation works to CPU clock speed.



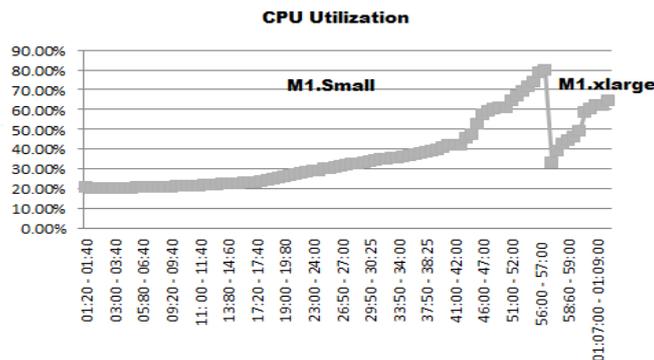
(Figure.3: Workload Pattern)



(Figure 4 . CPU Utilization of 2 different VMs types of same workload)



(Figure.5: Combine CPU Utilization of 2 instances with IAVGO)



(Figure.6: CPU Utilization with IAVGO)

Thus the above (figure.3) showing that IAVGO is act under workload pattern and it can be seen that workload used as dramatically increase/decrease every minute where workload volume is 140%. The only changes in the workload may be decreasing in the running time. However firstly, there was not clear trends. Most importantly, with in time (one minute) during the experiment increase/decrease happens above 100 times. Here, using the well-resourced VM one and only way proper way to handle the peaks of such workload. However, after that provides the resources in peak time so resources never waste.

On the other hand, if there is dynamic workload IAVGO maintains to achieve better instance time to complete the task by combining CPU utilization within green boundary. Thus the above (Figure. 4) shows combine CPU utilization data monitored at per minute basic system. The range goes on above 80% of dynamic workload. This illustrates that even for a VM which work under random workload that changes very frequently and according to optimization rule IAVGO still capable for retaining successors in appropriate VM size.

Figure.5 shows that combining M1.small with M1.xlarge when Threshold up time resources are utilized more, down limit varies. And threshold down time the resources are utilized less, up limit varies then CPU utilization contains the 80-85% .After all work is finish, using the automated rule it update in rule Meta data. So that rule Meta data only consider the peak time according to the CPU utilization. Finally the figure .6define the overall CPU utilization of each user VM size. To the algorithm if migrates size to user and starting point if any peak time allocate also size, so resources never waste.

VII. CONCLUSION AND FUTURE WORK

While several research proposes server and hardware control mechanism to achieve better efficient greener environment. In this paper argues another approach known as IAVGO, this optimization specifically for created IaaS VM users who usually have no control over any cloud hardware. We have demonstrated how IAVGO effectively facilitate automatic and intelligently VM allocation on discovering waste or overuse of VM resources. Experiments have shown that in this paper cloud always retain suitable successor and those who having pick time of green boundary migrate the resources, so that resources are utilized efficiently. The future work we address on I) Extended optimization based on additional metrics. II) Advanced load balancing techniques. III) Multiple services at peak time and CSP metadata support

REFERENCES

- [1] A. Younge, G. Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for Cloud Computing environments", *International Green Computing Conference*, pp.357-364, 2010
- [2] Amazon.com, "Amazon Web Services (AWS)," Online at <http://aws.amazon.com>, 2008.
- [3] www.wikipedia.com
- [4] [9] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "EnaCloud: An Energy-saving Application Live Placement Approach for Cloud Computing Environments", *IEEE International Conference on Cloud Computing*, pp. 17-24, 2009
- [5] Apache Software Foundation, "Apache JMeter", 2013, [Online], Available: <http://jmeter.apache.org/>, [Accessed: Mar. 15, 2013]
- [6] Q. Chen, P. Grosso, K. Veldt, C. Laat, R. Hofman, and H. Bal, "Profiling energy consumption of VMs for green Cloud Computing", *Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 768-775, 2011