ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Energy Efficient Control for Resources in Cloud Computing

Nivetha.S, Ram Kumar.A

M.E Student, Dept. of C.S.E., Arunai Engineering College, Thiruvannamalai, India

Asst. Professor, Dept. of C.S.E., Arunai Engineering College, Thiruvannamalai, India

ABSTRACT: Cloud computing is an emerging paradigm of remote server hosted on the internet to store, manage and process data rather than local server or personal computer. It provides various capabilities to store and process their data in the third-party. Three major services in cloud computing(Iaas, Paas and Saas).Iaas is a form of cloud computing, it will provides virtualized computing resources throughout the internet. It offers highly scalable resources that can be adjusted on demand. It makes Iaas well suited for the workloads that are temporary or changed unexpectedly. There are two major issues in cloud provider one is energy efficient and another one is performance guarantee. In cloud system, three policies are implemented for power saving. A control approach called N policy are implemented in which queuing system with N policy will turn a server on only when item in a queue is greater than or equal to predetermined N threshold limit. Rather than Activating a power off immediately, the N policy will results in performance degradation when a server stays in power saving mode too long. This leads to resource over provisioning, huge operational cost and high power consumption. In this paper open stack cloud provider is used to built the virtual instance. Open stack is cloud operating system that controls large pool of data to compute, storage and networking resources all managed through a dashboard. An energy efficient control in mitigating server ideal power has become a critical concern in designing a modern green cloud system. Virtual instance are created in open stack by using two ways, Virtual instance created directly in open stack dashboard, using jcloud API, instance created in open stack. A job is allocated to virtual instance in cloud. If a single virtual instance performs this job, power consumption and response time will be high. To overcome this issue, job splitter is used to divide the job in multiple small tasks by using improved efficient green control algorithm by using SR policy. Then the grid of virtual instances are allocated by each tasks to reduce response time and operational cost without sacrificing a SLA constraint.

KEYWORDS: OS, SLA, API, Suspend Resume Policy, Improved Efficient Green Control Algorithm

I. INTRODUCTION

A Cloud computing is an emerging service model for sharing a service and resources over the internet. cloud will provide three service Saas ,Paas ,Iaas. When the users become more dependent on cloud service and mobile platform increasing their ubiquitous of the cloud ,the quality of service become increasing important. The SLAS may be resource related ,performance related or even quality of service related information provides often finish up over provisioning their resources in order to meet the client's SLAS. Such over provisioning may increase the operational cost. Open stack is a cloud operating system that controls large pools of computer ,storage, and network resources throughout a data canter ,all managed through a dash board. An energy efficient control especially in mitigating server idle power has become a critical concern in designing a modern green cloud system. Generally a server operates alternately between a busy idea and sleep mode .A busy mode indicates that jobs are process by a server running in one or more of its ums where as idea mode defined as a server remains active but no job is being processed at a time sleep mode responsible for saving power consumption to eliminate or mitigate idea power wasted existing system uses three power saving policies with different energy efficient controls. Former polices is to make an energy efficient control in a system with three operation modes are used this policy will turn a server on only when item in a queue is greater than or equal to a predetermined N threshold instead of active a power off the server immediately upon an item arrival but it result in performance degradation .this is called N policy. Next policy is to reduce idea power consumption ,where it only holds two modes (busy ,sleep).instead of entering into an idle mode, a server immediately switches into a sleep mode when the system become empty ,if a no of job in a waiting queue is exceeds then switches into a busy mode .this

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

approach called as an SN POLICY. Last policy uses three operation mode. The server stays only in the sleep mode for a given amount of time ,the time expires then the server switches into an idle mode or to an busy mode depending upon the job arrival ,this is known depending upon the job arrival this is known as an SI POLICY(sleep to idle).In this paper open stack is used for cloud provider which is the best way to built virtual circumstance. A work is allocated to virtual Instance in cloud. If a single virtual instance perform this job, power consumption will e high and response time will be delayed. In order to provide energy efficient and low power consumption in cloud, job is divided into multiple small tasks based on the improved efficient green control algorithm with the help of an SR policy(Suspend resume policy) and the tasks are allocated to grid of virtual instances. Then the virtual instances performes job to reduce response time.

II. RELATED WORK

Power savings in cloud systems have been broadly studied on various aspects. e.g., on the virtual machine (VM) side by migrating VMs, applying allocation algorithms, and on the data center infrastructure side during resource allocations, energy managements, and so on…

Power-Saving in VM:

Huang et al. studied the virtual machine assignment problem with the aim of minimizing the total energy consumption. A multi-dimensional space division model and a virtual machine placement algorithm were presented. When a new VM assignment task arrived, their algorithm checked the subsequent resource usage state for each sufficient PM, and then chose the most suitable PM according to their replica to reduce the number of running PMs. In , Nathuji et al. considered the difficulty of providing power budgeting sustain while dealing with many problems that arise when budget virtualized systems. They managed power from the VM-centric point of view, where the goal was to be alert of global utility tradeoffs between different virtual machines when maintains the power constraints for physical hardware on which they can run. The approach to VM-ware power budgeting use multiple distributed managers integrated into the virtual power management framework. Yang et al. proposed an approximation algorithm and two dynamic programming to unite virtual machines . Two issues in energy conservation algorithm were addresses—the assignment of virtual machine representation and the characteristics of VM. In spite of that the dynamic programming could identify the best possible solution, time complexity was exorbitant in practice. In the energy competence from the performance perception was studied. Ye et al. presented a virtual machine based energy efficient data center system architecture for cloud computing. Then, they investigate the probable performance overheads caused by server consolidations and lived voyage of virtual machine technology. The potential performance overheads of server consolidation were evaluated.

Power-Saving in Computing Infrastructure:

Duggan and Young offered a basic theoretical model and used in building managing, micro-grids, and data center energy management. They analyzed these dissimilar energy management systems and defines a replica for resource allocation that could be used for these and other energy management system. The Data center Energy Management project was all ears on modelling energy consumption in data centers, with a target to optimize electricity consumption. Their project was focused on collect data to define basic fuel consumption curves .In Mazzucco et al. address the problem of improve the income of cloud providers by adornment down their electricity costs. Policies were based on dynamic estimates of user demand and system behavior models. Some approximations are used to handle the resulting models. They had demonstrated that decisions, such as how many server were powered on can have a note worthy effect on the revenue earned by the provider. However, no start up power draw or performance guarantees was considered. Zhang et al. presented Harmony, a dynamic Heterogeneity-aware Resource monitoring and management system that was able to perform dynamic capacity provisioning(DCP) in heterogeneous data centers. Using K means clustering, they show that the heterogeneous workload can be splitted into several task classes with similar characteristics in performance. The DCP originate an optimization problem that consider machine and workload heterogeneity as well as reconfiguration costs. A framework used to mechanically manage computing resources of cloud infrastructures was proposed to simultaneously achieve suitable Quality of service levels and to decrease the amount of energy used for providing services. Anglano and Canonico showed that via discrete-event system (DES) simulation, their solution was capable to manage physical resources of a data center to significantly reduce SLO violations corresponds to a traditional approach. The energy-efficiency of the infrastructure was defined as the amount of energy used to serve on its own application request. In, Amokrane et al. proposed Greenhead, a holistic resource management framework for embed virtual data across distributed data centers connected through a network. The goal of Greenhead was to capitalize on the cloud provider's

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

returns while ensuring that the communications was as environment-friendly as possible. They conducted extensive simulation of four data centers connected through the NSF Net topology. Choosing data center locations abounding by green energy sources could greatly shrink environmental pollution. The objectives were to optimize network performance,CO2 emissions, capital expenditure, and operational expenditures. The objective of cloud computing was to minimize the power consumption of the network. Their proposed model allowed planners to evaluate different solutions and to make variation in the optimization priorities. Although power management in cloud has attracted substantial research awareness, few studies focused on effectively reducing idle server power portray. Unlike previous studies, our paper contributes to investigate a necessary transaction between power consumption cost and system performance by applying different power-saving policies. To the best of our knowledge, applying the N-policy for optimizing the mode-switching control and simultaneously achieve the minimum cost under a performance guarantee has not been considered before.

III. PROPOSED SYSTEM

In this paper four modules are used to describe the system architecture. They are us follow

- OpenStack Configuration
- Build and Configure Instance
- Launch an Instance
- Power Management in Grid Computing

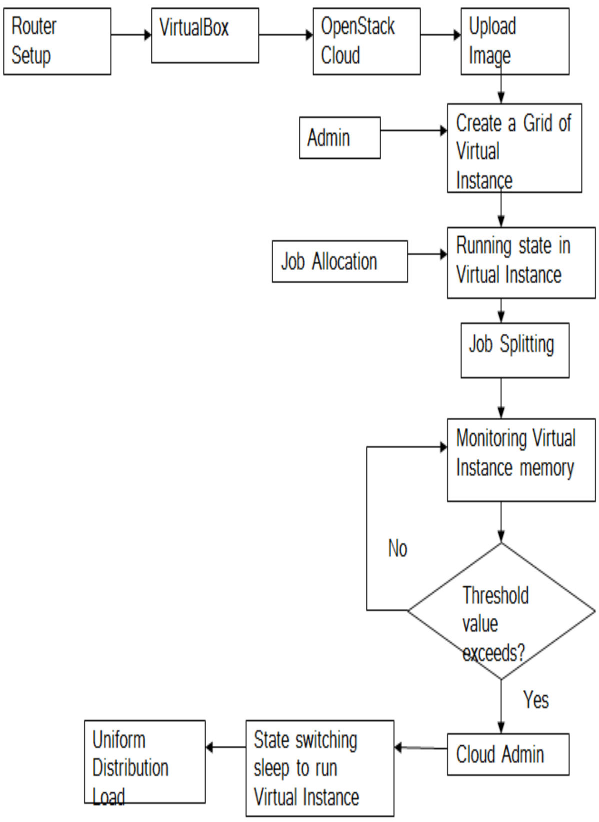

Fig 1. System Architcture

OpenStack Cloud Setup:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

In this module, Router configuration will be done. The following setting is done: Router IP address and WLAN settings. After Router setup, install VirtualBox. Disk space will be allocated to VirtualBox. Now import virtual appliance in VirtualBox. Next processor and adapter settings need to be set. Authentication step will be done to OpenStack. Server get started, it provides OpenStack IP and Authentication details in Terminal.

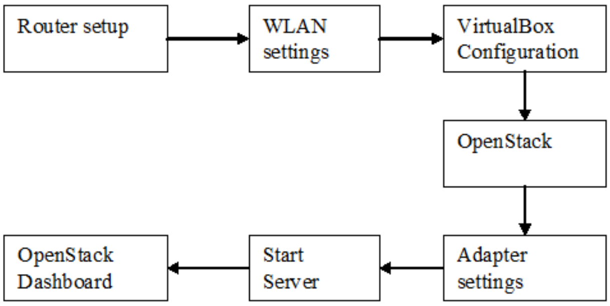

Fig1.a OpenstackCloudProvider

Build and Configure Instance:

In this module, Image and Instance will be created in OpenStack Dashboard. To open Dashboard, go to browser and type OpenStack IP. Dashboard gets opened. Provide username and password, the page gets redirected to OpenStack Admin Page. At the left side of Dashboard, under Admin Tab click Images link. In Image link, click Create Image button. A dialog box gets opened. Give name, description (optional), Select file in Image source, and upload an Image from system and choose format as QCOW2. Then click Public checkbox. Now click Create Image button. Image gets uploaded to OpenStack. To launch an Instance, the following steps to be follow: At the left side of the Dashboard, under Project Tab, click Instances link. Click Launch an Instance at the top right of Instance Page. Select Availability zone as nova, provide Instance name, flavor (RAM), Boot from Image as Instance Boot source and choose Image. Now click Launch button. A Virtual Instance get launched in OpenStack Dashboard

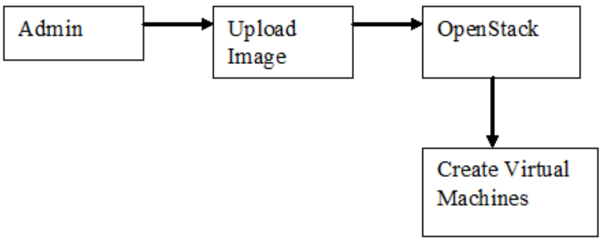

Fig1.b.Building and Configuring Instance

Grid Computing and Application Deployment:

In this module, Instance launched using jclouds API. To launch an Instance in OpenStack using jclouds API, provide authentication and openstack-nova provider to jclouds. Then provide RAM, image name and instance name. Using NodeMetaData, Instance will be launched in OpenStack Dashboard. A separate IP will be created for each virtual instance.

To construct a Grid of Virtual Instances, Admin has to provide total number of virtual instances to run, initial Virtual Instances to be in running state and RAM specification. Grid of Instances will be created one by one for the total number of Instances. The Minimum number of Instances to be run is retained and other Virtual Instances are put to sleep state. User can transfer a web application to Virtual Instance. Now user can deploy their web applications in

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Virtual Instance. Load balancing and SR policy is not implemented on Virtual Instances in Grid Computing which may lead to high overload and ultimately crashes the Server Instances. The power consumption and Response time will also be high.

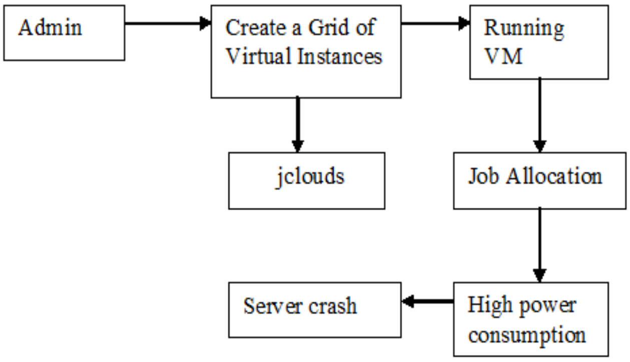

Fig1.c.Grid Computing and ApplicationNDeployment

Power Management in Grid Computing:

Admin will allocate Jobs to Virtual Instance. This job will put on FCFS queue and can be served in Grid Computing Environment. Job is split into small tasks and allocated to grid of running Virtual Instances. The memory for each Virtual Instance is monitored continuously to prevent overloading. A threshold value will be checked with memory usage and if any Virtual Instance exceeds, it will be reported to Cloud Admin. Load balancing is achieved by triggering the SR Policy to load another Virtual Instance which is in sleep state. This ensures uniform distribution load among all the Virtual Instances that helps in preventing high memory usage which will drastically influence power consumption.

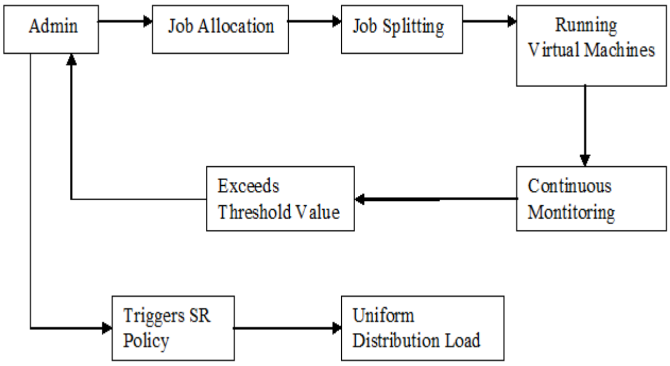

Fig1.d. Power Management in Grid Computing

IV. EXPERIMENTAL RESULTS

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Experiments are conducted to (i) validate that the optimal solution to minimize cost within a response time constraint can be obtained by applying the power-saving policies with the EGC algorithm. (ii) Cost reduction and response time improvement can be achieved at the optimal solution as compared to a general policy. Systems with different power-saving policies need to comply with the same response time guarantee.

The proposed power-saving policies combined with the EGC algorithm are evaluated on the basis of comparisons with a general policy. For a general policy, it implies that a solution
is given only by considering an absolute performance guarantee. Different arrival rates ranging from 360 to1,200 request/min are demonstrated to investigate a wide range of load intensities from an off-season to a peak-load period.

In order to compare the power-saving policies with a general policy, we mainly control the service rate to obey the same performance guarantee of SLA under various arrival rates. Comparisons of the idle probability in a system with the general policy, and sleep probabilities in systems with the power-saving policies are shown in Figure.

The sleep probabilities and the idle probability will be reduced as the value does not exist the threshold limit. If it exists the limit arrival rate increases since a server has more probability to work and stay in a busy mode. It is also noted that sleep probabilities are obvious larger than the idle probability in a system with the general policy due to the fact that our proposed algorithm tries to enhance efficiency in busy modes. Therefore, service rates are controlled at higher values with power-saving policies and their idle times can be reduced by switching into sleep modes. Conversely, the general policy focuses only on a performance guarantee and reduces the operational cost saving and response time improvement.

On the other hand, since a server switches into a sleep mode only once in an operation cycle for the SI policy, the variation among sleep probabilities is slight as the arrival rate increases the response time. The system with the SI policy doesn't reduce the sleep probability as the arrival rate increases; hence, it results in higher response times than other power-saving policies with a higher arrival rate. For the general policy, the response times are all kept at the 0.5 sec. Although the general policy tries to keep the service rate as low as possible, it still results in higher cost than other policies. Finally, we measure the cost improvement rates, which calculate the relative value of improvements to the original value instead of an absolute value.

It's noted that the proposed power-saving policies can effectively reduce cost, especially when an arrival rate is low. And also it improve the response time for a cloud provider who focuses on reducing cost, implementing the SR policy is a better choice to deal with a wide range of arrival rates.

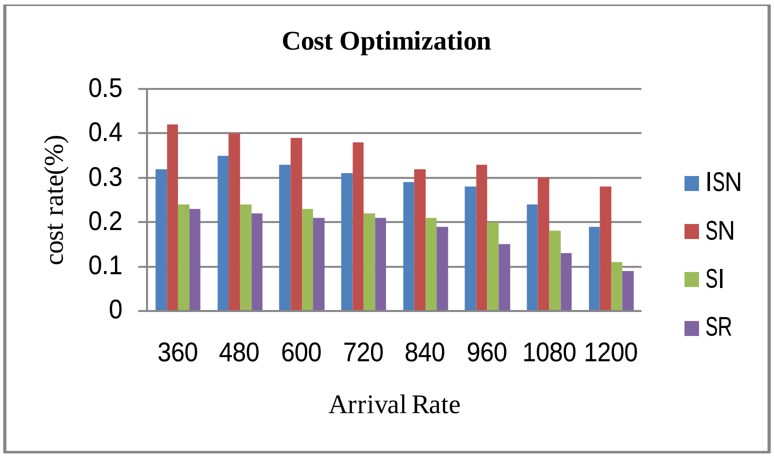

Fig2.a.Cost Optimization

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

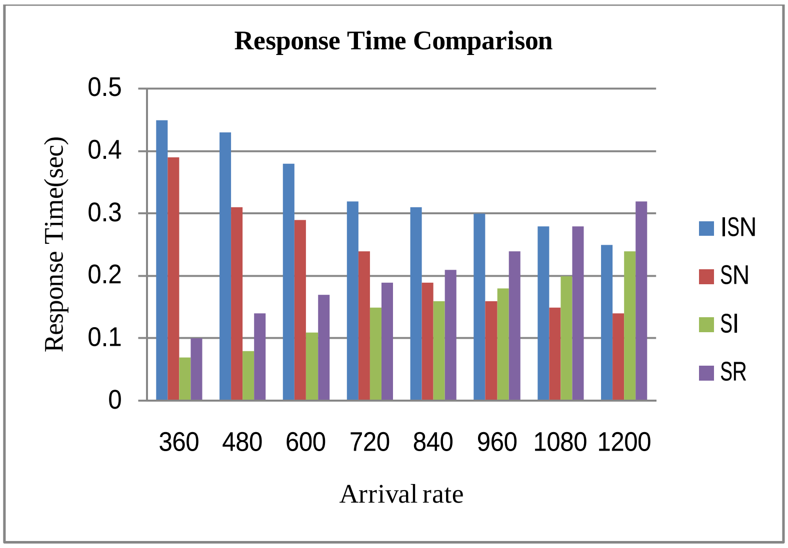

Fig2.b.Response time improvement

V. CONCLUSION AND FUTURE WORK

The growing crisis in power shortage has bought a concern in the future cloud system designs. To mitigate unnecessary ideal powers Suspend Resume (SR) policy are used with the efficient green control algorithm. Our proposed algorithm permit cloud providers to optimize the decision-making in service rate and mode-switching restriction, so as to minimize the operational cost without sacrificing a SLA constraint. The System uses open stack cloud provider to create the grid of virtual instance by using this energy efficiency is achived. Compared with other policy Cost savings and response time improvement can be achieved in SR policy.

REFERENCES

1. Yi-Ju Chiang,An Efficient Green ControlAlgorith In Cloud Computing For Cost Optimization, IEEE Transaction on Cloud Computing,vol . 3, no.2,2015.
2. Q. Zhang, et **al**, Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud, IEEE Transactions on Cloud Computing, vol. 2, no.1, pp.14-28, 2014.
3. J. Song, et al, " Study on energy-consumption regularities of cloud computing systems by a novel evaluation model," pp. 1-19, 2013.
4. A. C. Zhou, and B. He, "Transformation-based monetary cost optimizations for workflows in the cloud," IEEE Transactions on Cloud Computing, pp. 1-1, 2013.
5. P. Leitner, W. Hummer, and S. Dustdar, "Cost-based optimization of service compositions," IEEE Transactions on Services Computing, vol. 6, no. 2, pp. 239-251, 2013.
6. F. Larumbe, and B. Sanso, "A Tabu Search Algorithm for the Location of Data Centers and Software Components in Green Cloud Computing Networks," IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 22-35, 2013.
7. A. Amokrane, et al, "Greenhead: Virtual Data Center Embedding across Distributed Infrastructures," IEEE Transactions on Cloud Computing, vol. 1, no.1, pp. 36-49, 2013.
8. W. Huang, X. Li and Z. Qian, "An Energy Efficient Virtual Machine Placement Algorithm with Balanced Resource Utilization," Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 313-319, 2013.
9. Y. C. Lee, and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," The Journal of Supercomputing, vol. 60, no. 2, pp. 268-280, 2012.
10. R.Ranjan, R. Buyya, and M. Parashar, "Special section on autonomic cloud computing: technologies, services, and applications," Concurrency and Computation: Practice and Experience, vol. 24, no. 9, pp. 935-937, 2012.