International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 20 Issue 3 – MARCH 2016.

ENERGY EFFICIENT CONTROL FOR RESOURCES IN CLOUD COMPUTING

[1]Divya Sreenivasan, [2]K.S. Afrose Fathima, [3]S. Devi

[1,2]*UG Scholar, Computer Science and Engineering, Dhanalakshmi College of Engineering*
[3]*Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi College of Engineering*

[1]sreenivasandivya5@gmail.com
[2]faiyazfathi26@gmail.com
[3]devi.muthuvel@gmail.com

***Abstract* - OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a data centre, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface. OpenStack lets users deploy virtual machines and other instances which handle different tasks for managing a cloud environment. To satisfy uncertain workloads and to be highly available for users anywhere at any time, resource over-provisioning is a common situation in a cloud system. Existing system uses an N-policy and a Green control algorithm. This may result in performance degradation when a server stays in a power-saving mode too long under a larger controlled N value. Proposed system uses OpenStack Cloud provider which is the best to build Virtual Instance. A job is allocated to Virtual Instance in Cloud. If a single virtual instance perform this job, power consumption will be high and response time will be delayed. In order to provide energy efficiency and low power consumption in Cloud, Job is divided into multiple small tasks. These tasks are allocated to a grid of Virtual Instances that can split the task among the virtual instances that are in active state. When an overload of work happens, a virtual instance in suspend state is switched over to active/resume state, using an SR policy, to complete the job. Grid of Virtual Instances performs job to reduce response time.**

***Index Terms* – Energy-efficiency control, response time, power-saving policy.**

I. INTRODUCTION

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface. OpenStack lets users deploy virtual machines and other instances which handle different tasks for managing a cloud environment on the fly.

To satisfy uncertain workloads and to be highly available for users anywhere at any time, resource over-provisioning is a common situation in a cloud system. However, most electricity-dependent facilities will inevitably suffer from idle times or low utilization for some days or months since there usually have off-seasons caused by the nature of random arrivals. An energy efficient control, especially in mitigating server idle power has become a critical concern in designing a modern green cloud system.

To avoid switching too often, a control approach called N policy, had been extensively adopted. Queuing systems with the N policy will turn a server on only when items in a queue is greater than or equal to a predetermined N threshold, instead of activating a power-off server immediately upon an item arrival. However, it may result in performance degradation when a server stays in a power-saving mode too long under a larger controlled N value.

In this paper we use a variety for an efficient use of resources. An existing N-policy, defined by Yadin and Naor had been extensively adopted for the purpose of queuing. For energy-efficiency, ISN, SI and SN policy approaches have been used. In addition an improved efficient green control algorithm has been adopted which uses an additional state switching policy called SR policy for saving the energy consumed by idle instances.

We propose an energy efficient control for resources in cloud computing using which an allocated job can be efficiently performed in less time and without consumption of too much energy. A grid of virtual instances are used among which only some are in active mode. Others are suspended temporarily to save energy. The job is split and allocated to the active instances. When an overload or dump of work occurs, the suspended virtual instances are switched over to active state using a Suspend-Resume policy, and finish the job, thereby completing the work efficiently with less time, energy and resource consumptions.

The SI policy ensures that a job can get services without long delay, hence, it can achieve a lower cost than others when a system has a lower startup cost. Results show that to choose and implement the most suitable power-saving policy among diverse approaches, a cloud provider should take not only energy-efficient controls but also incurred costs into consideration since these have non-negligible impacts on system performances and operational costs. Here, our goal is to optimize the operational cost and of course, obey a response time guarantee via the optimal service rate and the controlled N value. An EGC algorithm is presented to solve the nonlinear constrained optimization problem effectively.

The main aim of this project is to efficiently utilize the idle power and to evenly distribute the load between the Virtual Instances thereby balancing memory utility and power factor, reducing operational costs.

II. EXISTING SYSTEM

A. N-POLICY

A control approach called N policy, defined by Yadin and Naor had been extensively adopted in a variety of fields, such as computer systems, communication networks, wireless multimedia, etc. Queuing systems with the N policy will turn a server on only when items in a queue is greater than or equal to a predetermined N threshold, instead of activating a power-off server immediately upon an item arrival. However, it may result in performance degradation when a server stays in a power-saving mode too long under a larger controlled N value. In this paper, the main contributions are summarized as follows.

1) Three power-saving policies that (a)switching a server alternately between idle and sleep modes,(b)allowing a server repeat sleep periods and (c) letting a server stay in a sleep mode only once in an operation cycle are all considered for comparison. The main objective is to mitigate or eliminate unnecessary idle power consumption without sacrificing performances.

2) The challenges of controlling the service rate and applying the N-policy to minimize power consumption and simultaneously meet a response time guarantee are first studied. To address the conflict issue between performances and power-saving, a tradeoff between power consumption cost and system congestion cost is conducted.

3) An efficient green control (EGC) algorithm is proposed to optimize the decision-making in service rates and mode-switching within a response time guarantee by solving constrained optimization problems. As compared to a typical system without applying the EGC algorithm, more cost-saving and response time improvements can be achieved.

B. ISN POLICY

Generally, a server operates alternately between a busy mode and an idle mode for a system with random job arrivals in a cloud environment. A busy mode indicates that jobs are processed by a server running in one or more of its VMs'; and an idle mode indicates that a server remains active but no job is being processed at that time. To mitigate or eliminate idle power wasted, three power-saving policies with different energy efficient controls, decision processes and operating modes are presented.

First, we try to make an energy-efficient control in a system with three operating modes m ¼ {Busy, Idle, Sleep}, where a sleep mode would be responsible for saving power consumption. A server is allowed to stay in an idle mode for a short time when there has no job in the system, rather than switch abruptly into a sleep mode right away when the system becomes empty [21]. An idle mode is the only operating mode that connects to a sleep mode. A server doesn't end its sleep mode even if a job has arrived; it begins to work only when the number of jobs in a queue is more than the controlled N value. According to the switching process (from Idle to Sleep) and the energy-efficient control (N policy), we have called such an approach the "ISN policy".

Although power is wasted in allowing a server to stay in the idle mode during a non-load period, the benefits are that an arrival job has more possibilities to get immediately service and the server startup cost can be reduced.

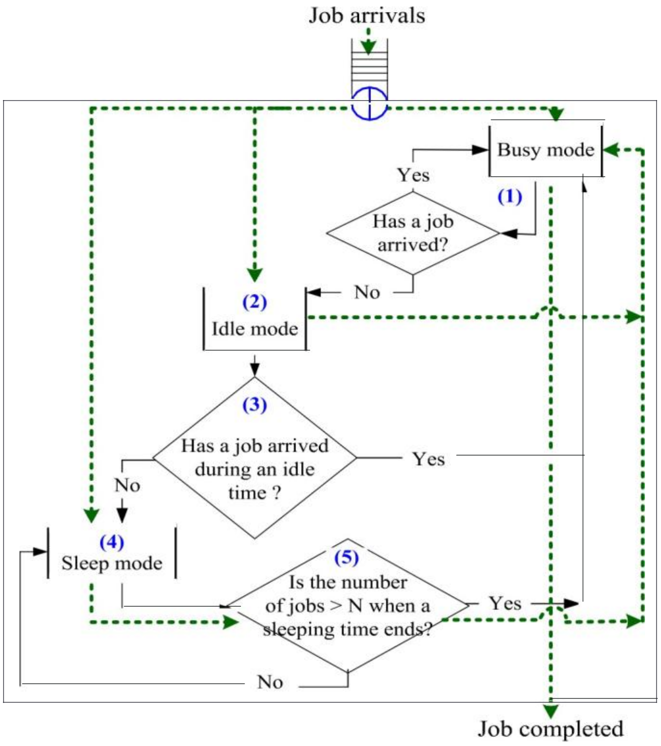

Fig. 1 Decision process of ISN policy

C. SN AND SI POLICIES

To greatly reduce idle power consumption, non-idle mode operating is considered in another approach, where it only holds {Busy, Sleep} operating modes. Instead of entering into an idle mode, a server immediately switches into a sleep mode when the system becomes empty. Similarly, a server switches into a busy mode depending on the number of jobs in a waiting queue to avoid switching too often.

The SN policy steps are as follows:

Step 1. A server switches into a sleep mode immediately when no job is in the system.

Step 2. A server stays in a sleep mode if the number of jobs in the queue is less than the N value; otherwise, a server switches into a busy mode and begins to work.

For comparison, the other policy is designed with no mode-switching restriction and performed under the other energy-efficient control. A server switches into a sleep mode right away rather than an idle mode when there has no job in the system.

This is similar to the SN policy but a server only stays in a sleep mode for a given time. When a sleeping time expires, it will enter into an idle mode or a busy mode depending upon whether a job has arrived or not. According to the switching process (from Sleep to Idle), we have called such an approach “SI policy”.

Step 1. A server switches into a sleep mode immediately instead of an idle mode when there has no job in the system.

Step 2. A server can stay in a sleep mode for a given time in an operation period. If there has no job arrival when a sleeping time expires, a server will enter into an idle mode. Otherwise, it switches into a busy mode without any restriction and begins to work.

III.PROPOSED SYSTEM

A. IMPROVED EFFICIENT GREEN CONTROL ALGORITHM

Proposed System implements Improved Energy-Efficient Green Control (EGC) Algorithm which uses SR policy (State Switching) in for improving energy-efficiency in Grid Computing. Proposed system uses Open Source Cloud Implementation, OpenStack to provide Cloud Architecture which is the best to build Virtual Instance. Virtual Instance created in OpenStack using two ways:

- Virtual Instance created directly in OpenStack Dashboard
- Using jclouds API, instance created in OpenStack

If a single job is allocated to one virtual instance at a time, power consumption will be high and response time will be delayed. In order to provide energy efficiency and low power consumption in Cloud, Job is divided into multiple small tasks. These tasks are allocated to a grid of Virtual Instances. Grid of Virtual Instances performs job to reduce response time. SR (Suspend Resume) Policy is implemented to provide efficient green control which allocates Virtual Instance based on the usage and can dynamically invoke the Virtual Instances on Demand. It also helps in balancing the load between the Virtual Instances as job arrives.

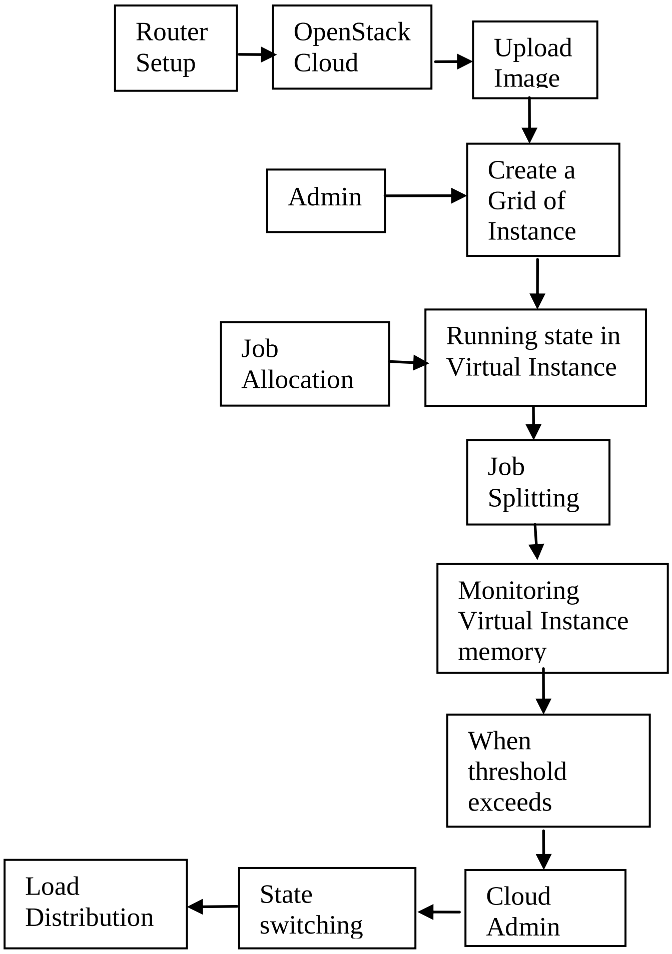

Fig. 2 System architecture

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. To start server, the following command to be used ./stack.sh. Server gets started, it provides OpenStack IP and Authentication details in Terminal. To stop server, the following command to be used ./unstack.sh.

Instances are 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 Node Meta Data, Instance will be launched in OpenStack Dashboard. A separate IP will be created for each virtual instance.

B. GRID COMPUTING AND APPLICATION DEPLOYMENT

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 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.

C. POWER MANAGEMENT IN GRID COMPUTING

The 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.

IV. CONCLUSIONS

The energy efficient control for resources in cloud computing, thus offers not only energy but also time, resources and job efficient system for cloud providers and provides ease of use to the cloud users.

Thus the energy efficiency is achieved in OpenStack cloud using Improved Energy-Efficient Green Control Algorithm and first come, first serve (FCFS) scheduling. Cost savings and response time improvement is verified.

REFERENCES

[1] G. Wang and T. E. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in Proc. IEEE Proc. INFOCOM, 2010, pp. 1–9.

[2] R. Ranjan, L. Zhao, X. Wu, A. Liu, A. Quiroz, and M. Parashar, "Peer-to-peer cloud provisioning: Service discovery and load-balancing," in Cloud Computing. London, U.K.: Springer, 2010, pp. 195–217.

[3] R. N. Calheiros, R. Ranjan, and R. Buyya, "Virtual machine provisioning based on analytical performance and QoS in cloud computing environments," in Proc. Int. Conf. Parallel Process., 2011, pp. 295–304.

[4] Server virtualization has stalled, despite the hype [Online]. Available: http://www.infoworld.com/print/146901, 2010.

[5] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," J. Supercomput., vol. 60, no. 2, pp. 268–280, 2012.

[6] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," Adv. Comput., vol. 82, pp. 47–111, 2011.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Comput. Syst., vol. 25, no. 6, pp. 599–616, 2009.

[8] L. Wang, G. Von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, "Cloud computing: A perspective study," New Generation Comput., vol. 28, no. 2, pp. 137–146, 2010.

[9] R. Ranjan, R. Buyya, and M. Parashar, "Special section on autonomic cloud computing: Technologies, services, and applications," Concurrency Comput.: Practice Exp., vol. 24, no. 9, pp. 935–937, 2012.

[10] M. Yadin and P. Naor, "Queueing systems with a removable service station," Operations Res., vol . 14, pp. 393–405, 1963.

[11] W. Huang, X. Li, and Z. Qian, "An energy efficient virtual machine placement algorithm with balanced resource utilization," in Proc. 7th Int. Conf. Innovative Mobile Internet Serv. Ubiquitous Comput., 2013, pp. 313–319.

[12] R. Nathuji, K. Schwan, A. Somani, and Y. Joshi, "VPM tokens: Virtual machine-aware power budgeting in datacenters," Cluster Comput., vol. 12, no. 2, pp. 189–203, 2009.

[13] J. S. Yang, P. Liu, and J. J. Wu, "Workload characteristics-aware virtual machine consolidation algorithms," in Proc. IEEE 4th Int. Conf. Cloud Comput. Technol. Sci., 2012, pp. 42–49.

[14] K. Ye, D. Huang, X. Jiang, H. Chen, and S. Wu, "Virtual machine based energy-efficient data center architecture for cloud computing: A performance perspective," in Proc. IEEE/ACM Int. Conf. Green Comput. Commun. Int. Conf. Cyber, Phys. Soc. Comput., 2010, pp.171–178. 154 IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 2, APRIL/JUNE 2015

[15] G. P. Duggan and P. M. Young, "A resource allocation model for energy management systems," in Proc. IEEE Int. Syst. Conf., 2012, pp. 1–3.

[16] M. Mazzucco, D. Dyachuky, and R. Detersy, "Maximizing Cloud Providers Revenues via Energy Aware Allocation Policies," in Proc. IEEE 3rd Int. Conf. Cloud Comput., 2010, pp. 131–138.

[17] Q. Zhang, M. Zhani, R. Boutaba, and J. Hellerstein, "Dynamic heterogeneity-aware resource provisioning in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 14–28, Jan.–Mar. 2014.

[18] M. Guazzone, C. Anglano and M. Canonico, "Energy-efficient resource management for cloud computing infrastructures," in Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci., 2011, pp. 424–431.

[19] A. Amokrane, M. Zhani, R. Langar, R. Boutaba, and G. Pujolle, "Greenhead: Virtual data center embedding across distributed infrastructures," IEEE Trans. Cloud Comput., vol. 1, no. 1, pp. 36–49, Jan.–Jun. 2013.

[20] F. Larumbe, and B. Sanso, "A tabu search algorithm for the location of data centers and software components in green cloud computing networks," IEEE Trans. Cloud Comput., vol. 1, no. 1, pp. 22– 35, Jan.–Jun. 2013.