ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 6, June 2016

ENERGY EFFICIENT CONTROL FOR RESOURCES OF A COLLABORATION ENVIRONMENT IN CLOUD COMPUTING

Prem Rosia. R[1], Vel Murugesh Kumar. N[2]
[1]PG Scholar,
Department of Computer Science and Engineering
Saveetha Engineering College, Tamil Nadu, India
9444683091

[2]Assistant Professor (SG),
Department of Computer Science and Engineering
Saveetha Engineering College, Tamil Nadu, India
9840850410

***Abstract*—The main objective in cloud collaboration environment is to save energy and optimize the cost. A computer system needs the highest energy with maximum utilization. Cloud computing is a new paradigm for delivering remote computing resources through a network. However, achieving an energy-efficiency control and simultaneously satisfying a performance guarantee have become critical issues for cloud providers. In this paper, three power-saving policies are implemented in cloud systems to mitigate server idle power. The challenges of controlling service rates and applying the N-policy to optimize operational cost within a performance guarantee are first studied. A cost function has been developed in which the costs of power consumption, system congestion and server start-up are all taken into consideration. The effect of energy-efficiency controls on response times, operating modes and incurred costs are all demonstrated. The results show that the benefits of reducing operational costs and improving response times can be verified by applying the power-saving policies combined with the proposed algorithm as compared to a typical system under a same performance guarantee.An efficient utilization ratio is less than 30% but the compute nodes consumes 46% while executing tasks. In past, we had an individual user/client to retrieve information from the single server. As nowadays, the resources are stored in the cloud and is used by multiple user/clients and it leads to demand access. Our objectives are to find the optimal service rate and mode-switching restriction, so as to minimize cost within a response time guarantee under varying arrival rates. An efficient green control (EGC) algorithm is first proposed for solving constrained optimization problems and making costs/performances trade-offs in systems with different power-saving policies.**

***Keywords*—Cloud Collaboration, Cloud Computing, Energy-Saving, Efficient Green Control (EGC), Power Consumption, Task scheduling.**

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.

In the existing system we had many disadvantages so that the system could not work as much as the user needed. The major issues are as follows: 1. Resource over-provisioning is that not much resources are loaded i.e. there will be a particular amount of resource it could store. 2. The working of all the machines leads to High power consumption. 3. Huge operational cost is due to the cost of buying the IT equipments. 4. Response time delay is caused by the overload of resources.

II. RELATED WORK

Presently, we have various knowledge, algorithms, and strategies nominated for improvement and negotiating energy in cloud computing systems, such as energy saving of compute nodes, energy-efficient

network, and power performance interchange and tackle the usable energy. Among these we had the vacation queuing model and task scheduling algorithms.

Task Scheduling algorithms are heuristic, meta-heuristic and queuing theory based. Let us see briefly about the task scheduling algorithms. Min-Min algorithm is the example for Heuristic scheduling here, the task is always assigned to the computer node with the least energy consumption. The advantage of the min-min algorithm is that it saves energy and there are two drawbacks 1.it ignores the performances of tasks, 2. Leads to poor adaptability.

The Greedy algorithm is the next example of heuristic scheduling. Here, the task execution time is assumed and is linearly related to energy consumption. Then transforms the energy parameters to time parameters (i.e.) normalization of energy and time. The drawback is that the stability is weak.

The higher level of heuristic is the meta-heuristic algorithm which has Genetic Algorithm (GA) with shadow price and Green task scheduling algorithm with shadow price Enhanced (SPGA). These two helps us to minimize task execution time with energy consumption constraint. Two measurements are used to find optimal solution and necessary randomness in the search process. 1. Fitness Value- evaluates the overall solution goodness, 2. Shadow Price- evaluates the component goodness.

In queuing-theory based algorithms the tasks are submitted to cloud computing system by the customers and the servers contains the heterogeneous compute nodes in the cloud computing system. Examples of queuing based are M/M/1, M/M/n, and M/G/1 models. These follow two strategies they are high utilization and low utilization task scheduling strategy for same purpose to reduce both idle and luxury energy. The advantage of this mode is that it analyses task completion in detail and drawback is that energy consumption is not considered.

We are familiar about three type of modes they are busy, idle and sleep modes. A control approach to avoid switching is carried out by authors Yadin and Naor (i.e.) the queuing systems with 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.A control approach called N Policy, had been extensively adopted in a variety of fields, such as computer systems, communication networks, wireless multimedia, etc. For energy-efficiency, ISN, SI and SN policy approaches used.

In this paper, we have many stochastic tasks running so we propose an energy- efficient green control algorithm which is used to solve optimization problems and cost/performance trade-offs.

III. PROPOSED SCHEME

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: 1.Virtual Instance created directly in OpenStack Dashboard, 2. 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.

IV. ADVANTAGES OF PROPOSED SCHEME

There are 4 main advantages in the proposed scheme they are: 1. Improved Energy Efficient Green Control, 2. Used jClouds API to create virtual instances, 3. Load Balancing and Scheduling, 4. Included Suspend Resume Policy.

V. ARCHITECTURE DIAGRAM

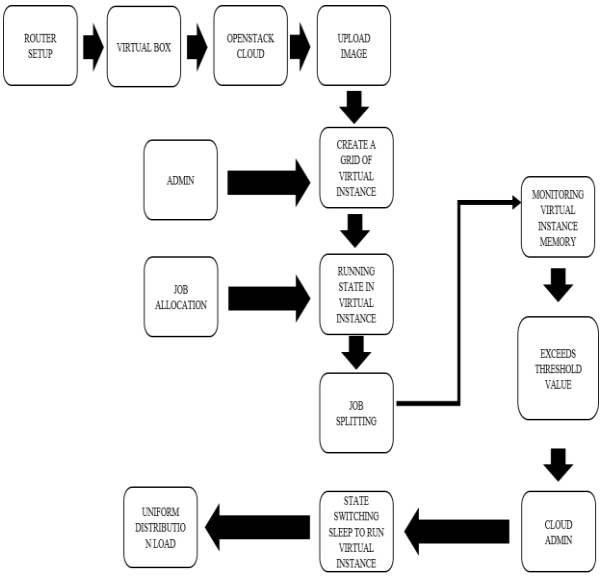

Fig 1. Architecture of Proposed System

The above architecture diagram illustrates about the specific structure of the whole system. It's the representation of the system and its behaviour. The flow process of each state is shown by arrows.

The admin create the virtual instances. Job is allocated by precedence and also it forms a queue mainly FCFS (First Come First Serve). Console are set to rum with different IP addresses. Each and every job is monitored. And job is splitted to the virtual machines to achieve load balancing. When 2 or more virtual machines is set to run, any one machine can be in sleep state. Because the other machines get overloaded it gets resumed automatically and runs the job.

The visible properties and the relationship between them are discussed in the forthcoming chapters. And also how the system progress through the entire network and the robustness to withstand the load and scheduling queues.

VI. SYSTEM IMPLEMENTATION AND MODULES

There are 4 modules implemented in this project which clearly project the whole system running in a green cloud which has a nature of storing and then sharing when it is needed.

A. OpenStack Setup

In this module, Router configuration will be done. We have to login the Router, the following setting is done: Router IP address and WLAN settings. After Router setup, install VirtualBox. A new window will be opened to import Disk Space. Browse for the folder to allocate the Disk Space. Disk space will be allocated to VirtualBox.

Now import virtual appliance in VirtualBox. Next system, processor and adapter settings need to be set. Authentication step will be done to OpenStack. To start server, login with user id and password, Change the directory by typing cd devstack. When you give ls command the list of files created will be shown.

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

B. 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, flavour (RAM), Boot from Image as Instance Boot source and choose Image. Now click Launch button. A Virtual Instance get launched in OpenStack Dashboard.

C. 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 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.

D. 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.

VII. CONCLUSION

Thus the energy efficiency is achieved in OpenStack cloud using Improved Energy-Efficient Green Control Algorithm and FCFS Scheduling. Experimental results shows that a system with the SR policy can significantly improve the response time in a low arrival rate situation. On the other hand, applying others policies can obtain more cost benefits when the start-up cost is high. As compared to a general policy, cost savings and response time improvement can be verified.

VIII. FUTURE ENHANCEMENTS

Speed of the virtual systems can be increased. Energy Saving is done by reduction of virtual servers by half, reducing power and cooling in half. Deployment of jobs can be maximized. EGC is used for both Corporate and Internet Data Centers. Lots of research to be carried out. Maximizing Efficiency of Green Data Centers. Most importantly the launching of many instances can be done according to the job scheduled.

REFERENCES

[1] Chunling Cheng, Jun Li and Ying Wang, (2015) An Energy – saving scheduling strategy based on vacation queuing theory in cloud computing, Tsinghua science and technology, ISSN volume 20.

[2] G. Lovasz, F. Niedermeier, and H. De-Meer, (2013) Performance trade-offs of energy-aware virtual machine consolidation, Journal of Networks Software Tools and Applications, vol.16, pp. 37–38.

[3] O. Philippe and L. Jorge, (2013) Deep network and service management for cloud computing and data centers: A report on CNSM 2012, Journal of Network and Systems Management, vol. 21, pp. 707–712.

[4]F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, (2014) iAware: Making live migration of virtual machines interference- aware in the cloud, IEEE Transactions on Computers, vol.63, pp. 3012–3025.

[5]J. Guo, F. Liu, D. Zeng, J. C. S. Liu, and H. Jin, (2013) cooperative game based allocation for sharing data center networks, in Proceedings IEEE Infocom, pp. 2139-2147.

[6] W. Deng, F. Liu, H. Jin, B. Li, and D. Li, (2014) Harnessing renewable energy in cloud datacenters: Opportunities and challenges, IEEE Network Magazine, vol. 28.

[7] **L**ei Rao, Xue Liu, Le Xie, Wenyu Liu, (2010) Minimizing Electricity Cost: Optimization of Distributed Internet Data Centres in a Multi-Electricity-Market Environment, IEEE Infocom.

[8] Young Choon Lee and Albert Y. Zomaya, (2011) Energy Conscious Scheduling for Distributed Computing Systems under Different Operating Conditions, Ieee Transactions On Parallel And Distributed Systems, Vol. 22, No. 8.

[9]Er. Shimpy and Mr. Jagandeep Sidhu, (2014) Different Scheduling Algorithms in Different Cloud Environment, International Journal of Advanced Research in Computer and Communication Engineering.

[10] A. Gandhi, M. Harchol-Balter, and A. M. Kozuch, (2012) Are sleep states effective in data centers? International Conference on Green Computing (IGCC), pp. 1–10.

[11] G. Daniel and T. Anthony, (2008) Cooperative load balancing in distributed systems, Concurrency and Computation Practice & Experience, vol. 20, pp.

[12] M. Y. Tan, S. G. Zeng, and W. Wang, (2012) Policy of energy optimal management for cloud computing platform with stochastic tasks, Journal of Software, vol. 23, pp. 266–278.

[13] S. Zikos and D. H. Karatza, (2011) Performance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times, Simulation Modeling Practice and Theory, vol. 1, pp. 239–250.

[14] L. Gong, H. X. Sun, and F. E. Waston, (2002) Performance modeling and prediction of non-dedicated network computing, IEEE Transactions on Computers, vol. 51, pp. 1041–1055.

[15]W. Wang, Z. J. Luo, and B. A. Song, (2013) Dynamic pricing based energy cost optimization in data center environment, Journal of Computers, vol. 36, pp. 600–615.

[16] Y. Z. Ma, (2006) The steady state theory of M/G/1 type multiple adaptive vacation queueing system, Ph. D. dissertation, Yanshan University, Qinhuangdao, China.

[17] M. Ware, K. Rajamani, M. Floyd, B. Brock, J. C. Rubio,F. Rawson, and J. B. Carter, (2012) Architecting for power management: The IBM POWER 7 approach, in Proc. IEEE 16th International Symposium on High Performance Computer Architecture, HPCA.

[18]H. Blume, V. J. Livonius, L. Rotenberg, T. G. Noll, H. Bothe, and J. Brakensiek, (2010) OpenMP-based parallelization on an MPcore multiprocess or platform—A performance and power analysis, Journal of Systems Architecture, vol. 54, pp. 1019–1029.

[19]X. Y. Shi, H. X. Jiang, and J. K. Ye, (2012) "An energy efficient scheme for cloud resource provisioning based on CloudSim", in IEEE International Conference on Cluster Computing, pp. 595–599.

[20]D. T. Braun and H. Siegel, (2012) A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous, Journal of the Parallel and Distributed Computing, vol. 61, pp. 810–837.

[21]F. Xiu, F. Liu, H. Jin, and A. V. Vasilakos, (2014) Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions, Proceedings of the IEEE, vol. 102, pp. 11–31.

[23]L. G. Valentini, W. Lassonde, U. S. Khan, N. Min- Allah, S. A. Madani, L. Juan, L. Zhang, L. Wang, N. Ghani, J. Kolodziej, et al., (2013) An overview of energy efficiency techniques in cluster computing systems, Cluster Computing, vol. 16, pp. 3–15.

[24]R. J, R. Kuo and C. W. Cheng, (2013) Hybrid meta-heuristic algorithm for job shop scheduling with due date time window and release time, The International Journal of Advanced Manufacturing Technology, vol. 67, pp. 59–71.

[25]G. Shen and Y. Zhang, (2013) Power consumption constrained task scheduling using enhanced genetic algorithms, in Evolutionary Based Solutions for Green Computing, Springer Berlin Heidelberg, pp. 139–159.

[26]L. X. Wang, P. Y. Wang, and H. Zhu, (2012) Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm, Mathematical Problems in Engineering, vol. 10, pp. 1–16.

[27]J. X. Chen, (2011) Energy efficient design of cloud data center, Cloud IDC, vol. 57, pp. 481–496.

[28]Y. S. Jing, A. Shahzad, and S. Kun, (2013) State-of-the-art research study for green cloud computing, Journal of Supercomputing, vol. 65, pp. 445–468.

[29]T. L. Chen and H. L. Lachlan, (2013) Simple and effective dynamic provisioning for power-proportional data centers, IEEE Transactions on Parallel and Distributed Systems, vol. 24, pp. 1161–1171.

[30]C. Y. Lee and A. Zomaya, (2012) Energy conscious scheduling for distributed computing systems under different operating conditions, IEEE Transactions on Parallel and Distributed Systems, vol. 22, pp. 1374–1381.

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

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

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

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

[35]R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, (2009) "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.

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

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

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