ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

# Active Resource and Job Allocation for Power Minimization in Mobile Cloud Systems

**Deepa.A [1], Ramkumar.S [2]**

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India
2. Asst.Professor, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

***Abstract*—Cloud computing allows business customers to scale up and down their resource usage based on needs. Many of the touted gains in the cloud model come from resource multiplexing through virtualization technology. In this paper, we present a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. We introduce the concept of "two level Scheme" to measure the unevenness in the multi-dimensional resource utilization of a server. We can combine different types of workloads nicely and improve the overall utilization of server resources. We develop a set of heuristics that prevent overload in the system effectively while saving energy used. Trace driven simulation and experiment results demonstrate that our algorithm achieves good performance.**

***Index Terms*—Cloud Computing, Resource Management, Virtualization, Green Computing.**

## I.INTRODUCTION

Cloud computing is the delivery of computing and storage capacity as a service to a community of end recipients. The name comes from the use of a cloud shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts services with a user's data, software and computation over a network. The remote accessibility enables us to access the cloud services from anywhere at any time. To gain the maximum degree of the above mentioned benefits, the services offered in terms of resources should be allocated optimally to the applications running in the cloud. The elasticity and the lack of upfront capital investment offered by cloud computing is appealing to any businesses. In this paper, we discuss how the cloud service provider can best multiplex the available virtual resources onto the physical hardware. This is important because much of the touted gains in the cloud model come from such multiplexing. Virtual Machine Monitors (VMMs) like Xen provide a mechanism for mapping Virtual Machines (VMs) to Physical Resources. This mapping is hidden from the cloud users. Users with the Amazon EC2 service [4], for example, do not know where their VM instances run. It is up to the Cloud Service Provider to make sure the underlying Physical Machines (PMs) has sufficient resources to meet their needs VM live migration technology makes it possible to change the mapping between VMs and PMs While applications are running [5], but, a policy issue remains as how to decide the mapping adaptively so that the resource demands of VMs are met while the number of PMs used is minimized. Christo Ananth et al. [3] discussed about a method, Optimality results are presented for an end-to-end inference approach to correct(i.e., diagnose and repair) probabilistic network faults at minimum expected cost. One motivating application of using this end-to-end inference approach is an externally managed overlay network, where we cannot directly access and monitor nodes that are independently operated by different administrative domains, but instead we must infer failures via end

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

to-end measurements. We show that first checking the node that is most likely faulty or has the least checking cost does not necessarily minimize the expected cost of correcting all faulty nodes. In view of this, we construct a potential function for identifying the candidate nodes, one of which should be first checked by an optimal strategy. Due to the difficulty of finding the best node from the set of candidate nodes, we propose several efficient heuristics that are suitable for correcting fault nodes in large-scale overlay networks. We show that the candidate node with the highest potential is actually the best node in at least 95% of time, and that checking first the candidate nodes can reduce the cost of correcting faulty nodes as compared to checking first the most likely faulty nodes.

Idle PMs can be turned off to save energy. In this paper, we presented the design and implementation of dynamic resource allocation in the Virtualized Cloud Environment which maintains the balance between the following two goals.

**Goals to Achieve:**

**Overload Avoidance :** The capacity of a PM must satisfy the resource needs from all VMs running on it. Or else, the PM is overloaded and leads to provide less performance of its VMs.

**Green computing :** The number of PMs used should be optimized as long as they could satisfy the need s of all VMs. And Idle PMs can be turned off to save energy. There is an in depth tradeoff between the two goals in the face of changing resource needs from all VMs. To avoid the overload, should keep the utilization of PMs low to reduce the possibility of overload in case the resource needs of VMs increase later. For green computing, should keep the utilization of PMs reasonably high to make efficiency in energy [7]. A VM Monitor manages and multiplexes access to the physical resources, maintaining isolation between VMs at all times. As the physical resources are virtualized, several VMs, each of which is self-contained with its own operating system, can execute on a physical machine (PM). The hypervisor, which arbitrates access to physical resources, can manipulate the extent of access to a resource (memory allocated or CPU allocated to a VM, etc.).

## II. LITERATURE SURVEY

Cloud systems with virtualization open a new opportunity to widen the scope of contention-aware scheduling, as virtual machines can cross legacy system boundaries with live migration. In this paper, we use live VM migration to dynamically schedule VMs for minimizing the contention on shared caches and memory controllers[4]. Furthermore, this study considers the effects of non-uniform memory accesses (NUMA) in multi socket systems commonly used in cloud servers. Contention-aware cloud scheduling techniques is used for cache sharing and NUMA affinity. The techniques identify the cache behaviors of VMs on-line, and dynamically migrate VMs, if the current placements of VMs are causing excessive shared cache conflicts or wrong NUMA affinity. Since the techniques identify the VM behaviors dynamically and resolve conflicts with live migration it will not required any prior knowledge on the behaviors of VMs[5]. The first technique, cache-aware cloud scheduling minimizes the overall last-level cache (LLC) misses in a cloud system. The second technique, NUMA aware cloud scheduling extends the first technique by considering NUMA affinity. Zhen Xiao [1] gives the strategy for dynamic resource allocation with Skewness and load prediction algorithm. He uses Xen hypervisor

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

Usher controller. The merits in his system are no overheads, high performance. It requires less number of migrations and residual resource is friendly to virtual machines. It improves the scheduling effectiveness. The demerit of the system is it is not cost effective. T.Wood [2] gives the Black and Grey box strategies with BG algorithm. He used Xen hypervisor and finds with Nucleous and monitoring engine, Grey-box enables proactive decision making. While it has the limitation as, Black-box is limited to reactive decision making and BG algo requires more number of migrations.

**III. EXISTING SYSTEM:**

In Existing system an algorithm for online scheduling of resources is proposed.There are two phases involved. The first phase is triggered when a task arrives for execution and second phase is triggered after the completion of the task. In the first phase, best fit is used to choose the server on which the task will run. In the second phase, all the tasks are sorted based on execution times, and a task that can run on the free Virtual Machine is chosen. This method performed better than all traditional methods such as bestfit, worstfit etc in terms of the CPU and memory utilization.

**Disadvantages**

- The same resource and VM would be over utilized, since ants would converge to first available position.
- Does not consider the mapping between VM's and hosts. I.e no load balancing is done.
- Longer jobs always tend to get pushed back, as shorter jobs get priority.
- It uses static allocation. Does not consider dynamic allocation

**IV. PROPOSED SYSTEM:**

The proposed system has a dynamic CPU/network resource and task allocation algorithm in cloud systems, called DREAM to minimize energy consumption of a mobile device for given delay constraints considering network traffic and offloadable/non-offloadable workloads in a unified framework. Tthe DREAM algorithm is first to jointly optimize cloud offloading policy and CPU/network speed scaling by invoking Lyapunov optimization technique in dynamic mobile network environment so as to answer howmuch energy can be saved further by the joint optimization.

- **Overload avoidance:** the capacity of a PM should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs.
- **Green computing:** the number of PMs used should be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save energy.

**Advantage of Proposed System:**

- We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.
- DREAM controls task (traffic/workload) type selection for CPU and networking resources
- DREAM controls clock speed scaling in CPU and interface selection in networking

**V. MODULE DESCRIPTION:**

**Number of Modules**

After careful analysis the system has been identified to have the following modules:

1. **Cloud Computing Module.**
2. **Resource Management Module.**
3. **Virtualization Module.**

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

### 4. Green Computing Module.

**Cloud Computing Module:**

Cloud computing refers to applications and services offered over the Internet. These services are offered from data centers all over the world, which collectively are referred to as the "cloud." Cloud computing is a movement away from applications needing to be installed on an individual's computer towards the applications being hosted online. Cloud resources are usually not only shared by multiple users but as well as dynamically re-allocated as per demand. This can work for allocating resources to users in different time zones.

**Resource Management Module:**

Dynamic resource management has become an active area of research in the Cloud Computing paradigm. Cost of resources varies significantly depending on configuration for using them. Hence efficient management of resources is of prime interest to both Cloud Providers and Cloud Users. The success of any cloud management software critically de-pends on the flexibility; scale and efficiency with which it can utilize the underlying hardware resources while pro-viding necessary performance isolation. Successful resource management solution for cloud environments, needs to provide a rich set of resource controls for better isolation, while doing initial placement and load balancing for efficient utilization of underlying resources.

**Virtualization Module:**

Virtualization, in computing, is the creation of a virtual (rather than actual) Version of something, such as a hardware platform, operating system, and a storage device or network resources.VM live migration is a widely used technique for dynamic resource allocation in a virtualized environment. The process of running two or more logical computer system so on one set of physical hardware. Dynamic placement of virtual servers to minimize SLA violations.

**GreenComputing Module:**

Many efforts have been made to curtail energy consumption. Hardware based approaches include novel thermal design for lower cooling power, or adopting power-proportional and low-power hardware. Dynamic Voltage and Frequency Scaling to adjust CPU power according to its load in data centers. Our work belongs to the category of pure-software low-cost Solutions. It requires that the desktop is virtualized with shared storage. Green computing ensures end user satisfaction, regulatory compliance, telecommuting, virtualization of server resources.

## VI. ALGORITHM USED IN PROPOSED SYSTEM

**A Two-Level Scheduling Algorithm for Energy efficiency:**

The algorithm follows a twolevel scheme of scheduling. In order to describe that, we define a set of tasks T = {t0, t1, t2, . . . tn1} and the number of tasks is n = | T |, and a set of hosts H = {h0, h1, h2, .13 . . hm1} and the number of hosts is m = | H |. The load on a host machine is defined as the average load of virtual machines that run on it.

We describe the algorithm below:

*Step 1:* Establish the host resource set, H = {h0, h1, h2, . . . hm1 }, and sort in ascending order of their processing power.

*Step 2* : Establish the set of tasks, T = {t0, t1, t2, . . . tn1 }. Then according to the properties of each task, the first level of the scheduler establishes the description of the virtual machine required, and thus providing configuration information for

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

allocation of resources and creation of the virtual machine.

*Step 3:* According to the virtual machine description of Task ti ∈T , select a host resource hj that can meet the required resources and the load is lightest. If the host exists, create the virtual machine and allocate the required resource for it then update the available resources Host hj, otherwise take the Task ti to the tail of the task queue and wait for the next scheduling.

*Step 4:* If the resource requirements of the Task ti increase, find whether the host whose virtual machine of Task ti run on can meet the additional required resources, if it exists, allocate the additional required resources for it, reconfigure the virtual machine, and then update the host's available resources.

Otherwise, the virtual machine is migrated to the host with lightest load and the additional required resources to execute continuously.

*Step 5* : If the resource requirements of the Task ti reduce, release the excess resources that the virtual machine occupied, and update the available resources held by the host.

*Step 6* : If Task ti has been completed, then destroy the virtual machine of Task ti and release the occupied resources for the other unfinished tasks.

*Step 7:* Calculate the load on each host, and find the standard deviation, if the load on a particular host is much higher than the rest, select a virtual machine with the lightest load from that host and migrate it to a host with the lightest load. If the load on a particular hosts is very light, migrate all VMs from it, to other hosts.

*Step 8* : Repeat Steps 3 to 7 until all tasks are completed.

In the above algorithm, the virtual machine is scheduled to the host with lightest load each time.

The advantage is to avoid overloading for the host. If a particular virtual machine is scheduled to a host and the computational amount increases, leading to a higher load on the virtual machine, resulting in load imbalance, then take the dynamic migration operation (described in Step 7), maintaining load balance in current environment.

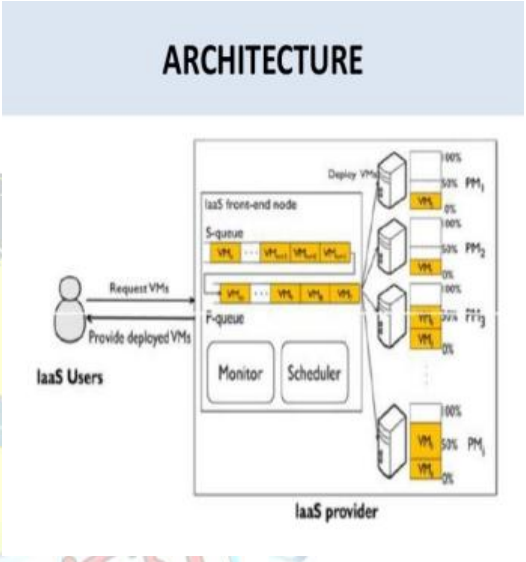


Fig.1 Architecture for Active Resource Allocation

## VII.CONCLUSION AND FUTURE ENHANCEMENT

We have evaluated the proposed twolevel task scheduling algorithm.The results of the experiments have shown that the proposed twolevel scheduling policy that does dynamic consolidation of virtual machines uses significantly less energy compared to other task scheduling algorithms that try to conserve energy like DVFS.We were able to achieve an energy reduction of 70 to 80%. Further in order to look into the cost of migrations, we recorded the SLA violations. We conclude that by migrating VM's with minimum migration time we achieve the least number of SLA violations. Such algorithms can be used by cloud IaaS providers to reduce power consumption, minimizing SLA

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

violations. In order to evaluate the proposed system in a real Cloud infrastructure, we plan to implement it by extending a realworld Cloud platform, such as OpenStack. Another direction for future research is the investigation of more complex workload models, e.g. models based on Markov chains, and development of algorithms that will leverage these workload models. Besides the reduction in infrastructure and ongoing operating costs, this work also has social significance as it decreases carbon dioxide footprints and energy consumption by modern IT infrastructures

**REFERENCES**

[1] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," technical report, Univ. of California, Berkeley, Feb. 2009.

[2] L. Siegele, "Let It Rise: A Special Report on Corporate IT," *The Economist,* vol. 389, pp. 3-16, Oct. 2008.

[3] Christo Ananth, Mona, Kamali, Kausalya, Muthulakshmi, P.Arthy, "Efficient Cost Correction of Faulty Overlay nodes", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 1,Issue 1, August 2015,pp:26-28

[4] "Amazon elastic compute cloud (Amazon EC2)," http://aws. amazon.comec2/, 2012.

[5] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," *Proc. Symp. Networked Systems Design and Implementation (NSDI '05),* May 2005.

[6] M. Nelson, B.-H. Lim, and G. Hutchins, "Fast Transparent Migration for Virtual Machines," *Proc. USENIX Ann. Technical Conf.,* 2005.

[7] M. McNett, D. Gupta, A. Vahdat, and G.M. Voelker, "Usher: An Extensible Framework for Managing Clusters of Virtual Machines," *Proc. Large Installation System Administration Conf. (LISA '07),* Nov. 2007.

[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-Box and Gray-Box Strategies for Virtual Machine Migration," *Proc. Symp. Networked Systems Design and Implementation (NSDI '07),* Apr. 2007.

[9] C.A. Waldspurger, "Memory Resource Management in VMware ESX Server," *Proc. Symp. Operating Systems Design and Implementation (OSDI '02),* Aug. 2002.

[10] G. Chen, H. Wenbo, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services," *Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '08),* Apr. 2008.

[11] P. Padala, K.-Y. Hou, K.G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant, "Automated Control of Multiple Virtualized Resources," *Proc. ACM European conf. Computer Systems (EuroSys '09),* 2009.

[12] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," *Proc. IFIP/IEEE Int'l Symp. Integrated Network Management (IM '07),* 2007.

[13] "TPC-W: Transaction Processing Performance Council," http://www.tpc.orgtpcw/, 2012. [14] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, and R.P. Doyle, "Managing Energy and Server Resources in Hosting Centers," *Proc. ACM Symp. Operating System Principles (SOSP '01),* Oct. 2001.

[15] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A Scalable Application Placement Controller for Enterprise Data Centers," *Proc. Int'l World Wide Web Conf. (WWW '07),* May 2007.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

[16] M. Zaharia, A. Konwinski, A.D. Joseph, R.H. Katz, and I. Stoica, "Improving MapReduce Performance in Heterogeneous Environments," *Proc. Symp. Operating Systems Design and Implementation (OSDI '08),* 2008.

[17] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair Scheduling for Distributed Computing Clusters," *Proc. ACM Symp. Operating System Principles (SOSP '09),* Oct. 2009.

[18] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling," *Proc. European Conf. Computer Systems (EuroSys '10),* 2010.

[19] T. Sandholm and K. Lai, "Mapreduce Optimization Using Regulated Dynamic Prioritization," *Proc. Int'l Joint Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '09),* 2009.

[20] A. Singh, M. Korupolu, and D. Mohapatra, "Server-Storage Virtualization: Integration and Load Balancing in Data Centers," *Proc. ACM/IEEE Conf. Supercomputing,* 2008.

[21] Y. Toyoda, "A Simplified Algorithm for Obtaining Approximate Solutions to Zero-One Programming Problems," *Management Science,* vol. 21, pp. 1417-1427, Aug. 1975.

[22] R. Nathuji and K. Schwan, "Virtualpower: Coordinated Power Management in Virtualized Enterprise Systems," *Proc. ACM SIGOPS Symp. Operating Systems Principles (SOSP '07),* 2007.

[23] D. Meisner, B.T. Gold, and T.F. Wenisch, "Powernap: Eliminating Server Idle Power," *Proc. Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS '09),* 2009.

[24] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, "Somniloquy: Augmenting Network Interfaces to Reduce Pc Energy Usage," *Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI '09),* 2009.

[25] T. Das, P. Padala, V.N. Padmanabhan, R. Ramjee, and K.G. Shin, "Litegreen: Saving Energy in Networked Desktops Using Virtualization," *Proc. USENIX Ann. Technical Conf.,* 2010.