An Efficient Virtual Machine Management to Achieve Energy Efficiency in Cloud Computing

Divya H. Rami[1], Riddhi V. Mehta[2]

Assistant Professor, AI&DS Department, Sankalchand Patel College of Engineering, Visnagar, India[1]
Assistant Professor, ICT Department, Sankalchand Patel College of Engineering, Visnagar, India[2]

divyarami14@gmail.com[1], mehta.riddhi85@gmail.com[2]

Abstract: Cloud computing has really changed the game in the world. It allows users to get computing resources whenever they need them. The demand for power has soared because of scientific, business, & web applications. This created huge data centers that use a LOT of electricity. So, how can we use this energy wisely? How do we become more energy efficient? These questions are super important in Green Cloud Computing. This study shares a new technique & algorithm designed for Efficient Virtual Machine Management to help save energy. Good Virtual Machine Management starts with how VMs are allocated. Throughout this research, there are four big steps: detecting overloaded hosts, choosing the right VMs, placing them correctly, & finally, spotting underloaded hosts.

Keywords: Virtual Machine Management, Virtual Machine, Cloud Computing, Virtualization, Power Consumption

---

I. INTRODUCTION

Cloud computing is like a magic portal! It lets people quickly get access to shared stuff, such as servers, storage, and applications, right from the internet. It's super convenient. You can set it up and take it down easily without needing much help or extra management.

The main idea behind cloud computing is to shift work from personal computers to powerful servers. This way, everyone can use the same hardware and software together. Yay! No more wasted resources on individual machines. Plus, it makes everything run more efficiently [1]. However, with so many people using the cloud now, all those big data centers need a lot of power. Whoa! That’s a lot of energy! This high demand for power clashes with today’s goals of saving energy & reducing carbon footprints – which is a BIG issue we can’t ignore. So, this paper aims to find a balance between advancing cloud tech & improving energy use. We can't just overlook these energy issues in the cloud world! To save energy, we can manage Virtual Machines (VMs) better. If a server has too many tasks (like being overloaded), we can shift some VMs around. Then, if a server isn’t busy enough (underloaded), we might turn it off completely to save power[1].

In our study, we came up with a neat system for keeping things running efficiently by managing VMs based on how much power each host uses. First, we figure out how much power every host consumes. Then we choose which one has the least usage and send VMs there for a smoother operation! Our method is all about helping with energy-saving & VM management by looking at CPU use and memory along with power consumption. Plus, we'll even use special algorithms to help virtual machines move around in order to save even more energy! In the end, our experiments show that this fun approach can really save energy in regular-use situations!

II. BACKGROUND

A. Cloud Computing

It is basically using the Internet to do computing. It lets you share hardware, software, & messages whenever you need them on computers or other gadgets [2]. Think of the cloud as a big metaphor for the network—the Internet! Users don’t have to dive into the details of how this “cloud” works. No need for deep knowledge or hands-on control here. Cloud computing helps businesses launch applications super-fast and simplify management [2]. Plus, it cuts down costs so companies can quickly shift IT resources based on what they need.

http://doi.org/10.63766/spujstmr.24.000012

You can think of cloud computing as having three main service levels: Infrastructure as a Service, which we call IaaS, Platform as a Service (PaaS), and Software as a Service (SaaS). It's all about making technology easier and more flexible so everyone can benefit!

B. Virtualization

Virtualization has become super important in the cloud computing world. It lets us run many Virtual Machines on a single Physical Machine while giving each Virtual Machine its own private space [3]. The software that helps manage this setup is called a virtual machine monitor or hypervisor. Popek and Goldberg first described a Virtual Machine as a smart, isolated copy of a real Machine. This setup makes it easier to share the resources of the Physical Machine. Thanks to virtualization, we can finely tune how we distribute resources to each Virtual Machine.

With these technologies, we can run multiple operating system instances—those are just fancy ways to say "Virtual Machines"—on one piece of hardware. Every VM works just like a real Physical Machine, acting as if it has its own Operating System and software running on it! Each VM needs to manage its access controls, which can be different from one VM to another, even when they're sharing the same hardware [3].

Some virtualization platforms need an extra host operating system, while others are built right into the hardware itself. There are several common methods for virtualization [4]. The biggest difference lies in which part keeps an eye on and controls the Virtual Machines. Sometimes it's the hosting operating system; other times, it's the special privileges assigned.

C. VM Management

We required a virtual machine to set up a standard cloud service. When you try to run lots of virtual machines using virtualization technology, things can get really tricky to manage. That's why we need a handy platform to manage all those virtual machines!

This management platform does a lot of cool things. It creates, edits, switches, pauses even does live migrations. Isn't that great? There are some popular open-source virtualization management platforms out there. They act like the network interface, making it super easy to build things virtually and enjoy many benefits [5]. For example, these platforms provide a nice interface to monitor the status of so many virtual machines at once! Plus, managing account permissions becomes simpler too. It's pretty neat how all these things come together!

III. METHODOLOGY

This study aims to find a way to use energy more efficiently. So, we set up a virtual machine cluster. This cluster is made up of physical machines that act as hosts.

To manage virtual machines well, we follow these steps:

Step 1: First, we figure out the total resource weight of our virtual machine cluster. This means adding up the resource weight of every single virtual machine. Then, we divide that by the total resource weight available on the host or machine. To get this total acquired resource weight, we use this equation (1):

$$\mathrm{HOSTjrate} = \sum_{i=1}^{v} \mathrm{VM}_{jirate} \tag{1}$$

$$\mathrm{VM}_{jirate} = \frac{\mathrm{VM}_{ji\mathrm{CPUuse}} \times \mathrm{VM}_{ji\mathrm{RAMallocate}}}{\sum_{i=1}^{n}(\mathrm{VM}_{ji\mathrm{CPUuse}} \times \mathrm{VM}_{ji\mathrm{RAMallocate}})} \tag{2}$$

Where j marks the serial number for each physical machine, I represent the serial number of the virtual machines. N stands for the total number of virtual machines. VMjiCPUuse shows how much processing power a virtual machine uses, while VMjiRAMallocate is all about how much memory is allocated to a virtual machine in its host.

Let's move to 2! First, we want to figure out the resource weight of the host (or physical machine). Check which host has the most resource allocation—that's our surcharged host.

Next, in Step 3, you'll want to look at VMs from that surcharged host. Pick the VM based on how quickly it migrates. The one that takes the least time? Yep, that's your choice for migration!

Now, for Step 4, it's time to find out where this VM should go. You'll need to check how much power each host uses. Choose the one consuming the least power! Look for the smallest difference in power consumption before and after migrating. The host with that tiny gap? That's where you'll relocate your chosen VM.

To reach our goal, we need to empty out the under-loaded hosts first. Only after that can we shut them down and get some rest for them!

So, in Step 5, let's find those under-loaded hosts with the least resource weight.

Step 6 involves selecting all VMs from these under-loaded hosts.

Then, in Step 7, just follow our earlier method for VM placement to put those selected VMs where they belong.

Finally, in Step 8, we've got an under-loaded host that's all set with an empty pipeline! So now we can turn off this host—it helps save energy!

IV. ALGORITHM

- For every host in the hostlist, we start like this: Input is Hostlist & Vmlist. We calculate (HOSTjrate - α). Now, if the host is overloaded, we add the VM with the shortest migration time to VMs To Migrate. This comes from the overloaded host.
- Then, we put it into a migrationMap to get a new VM placement for those VMs we want to move. Next up!
- Loop through each host in the hostlist again. Start with minPower set to MAX and allocatedHost to NULL. Check each host in the list. Does it have enough resources for the VM?
- If it doesn't get overloaded after we move the VM, we estimate Power for that host and VM combo. If our estimated power is less than minPower, then it's a match! We update the allocatedHost to that host & set minPower to that new power level.
- If allocatedHost isn't NULL after that, we add our VM allocation to that allocatedHost. Finally, we return our allocation.
- Now, let's check each host again! If a host is underloaded, we collect its VMs in VMsToMigrate and also add its new VM placement in the migrationMap. And that's what we return at the end!

V. DISCUSSION

In this research, there are four key stages: (1) Finding an overloaded host, (2) VM Selection, (3) VM Placement, & (4) Finding an underloaded host.

A. Finding Overloaded Host:

To manage VMs and improve energy efficiency in their management, we start by identifying any overloaded hosts. This step relies on setting upper & lower limits for resource use. The goal is to keep CPU usage among all VMs within these boundaries. If a host's CPU drops below the lower limit, all VMs must be moved to a different host. Basically, we put the idle one to sleep to save power. On the other hand, if usage goes above the upper limit, some VMs need to be migrated to avoid breaking any service level agreements (SLAs). We call the host with the highest resource usage an overloaded host. The more resources it's maxed out on, the higher its CPU utilization is too.

B. VM Selection:

Once we find an overloaded host, the next step is picking a new home for VMs that need relocating. When we decide that a host is overloaded, we look for specific VMs to move from that host. After choosing a VM to migrate, we check again to see if the original host is still overloaded. If it is, we repeat the selection process until it isn't considered overloaded anymore. There are multiple factors that influence which VM gets selected—these include low utilization rates or even random choices based on CPU demands.

C. VM Placement:

Next up after selecting VMs is deciding where they will land; this is called VM placement. Different strategies help us choose the best new home based on several factors like power use & cost efficiency. We want the spot with the least power consumption for placing these VMs when possible.

D. Finding Underloaded Hosts:

To find underloaded hosts, we take a simple approach. First, we assign chosen VMs to their new hosts after locating all overloaded ones using our detection method. Then, we try our best to spread those VMs out from the source host with the lowest usage so other hosts don't get too crowded. If this works well, those VMs will migrate successfully to their new homes, & once it's all done, that source host can go into sleep mode! If moving all the VMs isn't possible, then that source host stays active instead—as a result, this loop continues for every host until balanced.

E. Implementation & Testing Environment:

(a) Implementation Details:

We used Python for our proposed algorithm and made use of libraries like NumPy for calculations and pandas for organizing data. The simulation setup included:

1. Virtualization Platform: VMware vSphere 7.0
2. Cloud Management Tool: OpenStack 20.0
3. Power Consumption Modeling: Custom scripts were crafted in Python to estimate power needs based on how busy each host was.

(b) Testing Environment:

Our experiments took place in a controlled data center filled with:

1. Physical Hosts: 10 servers boasting 16 CPU cores plus 64 GB RAM each!
2. VMs: 100 virtual machines containing various CPU & RAM setups
3. Workloads: We simulated realistic workloads mimicking everyday application demands
4. Monitoring Tools: Nagios helped us keep an eye on performance while PowerAPI estimated energy consumption

(c) Testing Procedure:

1. Baseline Measurement: First off, we recorded power use and resource levels before any migration happened.
2. Algorithm Execution: Next up came applying our VM management algorithm while moving VMs based on our rules.
3. Post-Migration Measurement: Then came measuring power and resource use after everything was done.
4. Analysis: Finally, comparing results pre- and post-migration let us see energy savings plus performance differences.

F. Results & Comparative Analysis:

The current theory looks at how overcrowded physical machines (PMs) get identified and how then a VM moves from them to another PM that has lower usage rates after migration takes place! There's a table outlining this relation—showing how at first, we target the overloaded PM as a source and send its VM off to one that's less busy! Table number two illustrates how such migrations tie in with power consumption too! Power usage stood out as key during migrations; higher usages prompted migration much more often from those PMs!

This friendly analysis highlights how our method can help balance loads efficiently while saving energy—making everything run smoother overall

TABLE I:
VM OCCUPYING RESOURCE WEIGHT AFTER MIGRATION [10]

PM	VM	VMjicpuuse	VMjiramallocate	VMjirate	HOSTjirate
11	VM01	95	512	0.08	0.28
	VM03	40	2048	0.14	
	VM04	10	512	0.01	
	VM05	30	1024	0.05	
12	VM06	70	1024	0.12	0.17
	VM07	60	512	0.05	
13	VM02	100	1024	0.17	0.22
	VM08	10	1024	0.02	
	VM09	15	512	0.01	
	VM10	20	512	0.02	
14	VM11	45	1024	0.08	0.33
	VM12	60	1024	0.05	
	VM13	30	512	0.03	
	VM14	100	512	0.17	

Now the example according to the proposed work is given in Table 2: VM migration considering the power consumption.

TABLE II:
VM MIGRATION CONSIDERING THE POWER CONSUMPTION

PM	VM	VMjirate	HOSTjirate	Power Consumption
11	VM01 VM03 VM04 VM05	0.08 0.14 0.01 0.05	0.28	100.75
12	VM06 VM07	0.12 0.05	0.17	9.31
13	VM02 VM08 VM09 VM10	0.17 0.02 0.01 0.02	0.22	87.7

VI. CONCLUSION

Choosing how to manage virtual machines can really help save energy. This means less power use, which is great for cloud service providers, too! In our project, we focus on electricity as the main thing to look at. There's a theory that talks about migration based on host rates. But we're doing it differently—using power consumption to move virtual machines around and decide where they should go. After we migrate, the virtual machine will go onto the physical machine that uses the least power. Our goal? To get power usage as low as possible! We think this will improve energy efficiency—a lot! All in all, our research could make a real positive impact on people, unlike what's been done before. The algorithm we're suggesting shows that managing VMs can really cut down on energy waste. This can lead to less energy used, more profits for cloud providers, and lower costs for users. Best of all? A greener cloud computing future! However, just so you know, this idea hasn't been tried out or tested yet.

REFERENCES

[1] Hua, Wei, Peng Liu, and Linyu Huang. "*Energy-efficient resource allocation for heterogeneous edge-cloud computing*." *IEEE Internet of Things Journal*, 2023.

[2] Ajmera, Kashav, and Tribhuwan Kumar Tewari. "*Dynamic Virtual Machine Scheduling Using Residual Optimum Power-Efficiency In The Cloud Data Center*." *The Computer Journal* 67.3,2024, 1099-1110.

[3] National Institute of Standards and Technology. *The NIST definition of cloud computing*. http://www.nist.gov/itl/cloud/

[4] R. Buyya, C. 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 Computer Systems, 2009, 25 (6) pp. 599– 616.

[5] Abaslom Et-Shamir Ezugwu, Sahalu Junaidu, Seyed Buhari, "W*Virtual Machine Allocation in Cloud Computing" International Journal of Cloud Applications and Computing*, 3(2), 47-60, April-June 2013

[6] Shin-ichi Kuribayashi "*Reducing Total Power Consumption Method Include Computing Environments*" International Journal of Computer Networks & Communications (IJCNC) Vol.4, No.2, March 2012

[7] Harman Preet Kaur, Jasmeet Singh Gurm, "*A Survey on the Power and Energy Consumption of Cloud Computing*" International Journal of Computer Science Trends and Technology (IJCST) – Volume 3 Issue 3, May-June 2015

[8] Rakesh Kumar Vishwarkarma1, Dr. Syed Imran Ali2, Anidra Katiyar "*Energy Efficient Dynamic Resource Scheduling for Cloud Data Center*" International Journal of Recent Development in Engineering and Technology (ISSN 2347-6435(Online) Volume 5, Issue 7, July 2016)

[9] Yen, J.-L Yang, M.-J- Yang, Chao-Tung, "*Power-Saving Management for Energy-Efficient Green Clouds*" 2014/05/01, 381, 389, Journal of Internet Technology, 15 10.6138/JIT.2014.15.3.07

[10] Olayinka Adeleye *"Energy Efficient Virtual Machine Management for Cloud Computing: A Survey"* Department of Computer Engineering Federal University Oye-Ekiti, Nigeria International Journal of Scientific & Engineering Research, Volume 6, Issue 11, November 2015

[11] Anton Beloglazov and Rajkumar Buyya *"Energy Efficient Resource Management in Virtualized Cloud Data*

Centers" cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia

[12] Chao-Tung Yang∗, Jung-Chun Liu, Kuan-Lung Huang, Fuu-Cheng Jiang "*A method for managing green power of a virtual machine clustering cloud"* Department of Computer Science, Tunghai University, Taichung 40704, Taiwan Future Generation Computer Systems 37 (2014) 26–36

[13] Anton Benglov and Rajkumar Bhuya, *"Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers"* DOI: 10.1002/cpe.1867

[14] Rodrigo N. Calheiros1, Rajiv Ranjan2, Anton Beloglazov1, Cesar A. F. De Rose ´ 3 and Rajkumar Buyya1. "*CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms*" Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia 2011; 41:23–50

[15] T.Swathi, K.Srikanth, S. Raghunath Reddy, "*Virtualization in Cloud Computing"* IJCSMC, Vol. 3, Issue. 5, May 2014, pg.540 – 546

[16] Erol Gelenbe, Marco Di Girolama, Giovanni Giuliani, Andreas Beri, "*Energy-Efficient Cloud Computing*," Oxford University Press on behalf of The British Computer Society, 28 July 2009

[17] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "*A taxonomy and survey of energy-efficient data centers and Cloud computing systems*," Advances in Computers, M. Zelkowitz (ed.), vol. 82, pp. 47–111, 2011.

[18] Mohsen Sharifi, Hadi Salimi, Mahsa Najafzadeh, "*Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques*," Springer Science Business Media LLC(2011), 2011.

[19] M.Blackburn, "*Five ways to reduce data center server power consumption*: *Five ways to save server powe*r", the green grid. http://www.thegreengrid.org/ VMware Distributed power Management (DPM) http://www.vmware

[20] A. Wierman, L. L. Andrew, and A. Tang, "*Power-aware speed scaling in processor sharing systems*," in Proceedings of the 28th Conference on Computer Communications (INFOCOM), 2009, pp. 2007–201

[21] Minas L, Ellison B. Energy Efficiency for Information Technology: "*How to Reduce Power Consumption in Servers and Data Centers*". Intel Press, 2009.

[22] Beloglazov A, Buyya R. "*Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers*". Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, Bangalore, India, 2010