International Journal of Advances in
Soft Computing and Intelligent Systems (IJASCIS)
2023, Vol 02, Issue 02, 22-30

Science Transactions © 2023

Original Paper

A LOAD BALANCE AND VIRTUAL MACHINES MIGRATION APPROACH USING QEMU-KVM FOR ENERGY EFFICIENT DATA CENTRE OF CLOUD COMPUTING

Dr. Rajesh Patel[a] *, Dr.Kirit Modi[b] and Prof. Mehul Patel[c]
[abc]Sankalchand Patel University ,Visnagar, 384315, Gujarat, India
drrppatel.ce@spcevng.ac.in, kjmodi.ce@spcevng.ac.in , mspatel.it@spcevng.ac.in

Abstract

Load balancing and virtual machine migration are particularly challenging jobs in a cloud data centre. Some virtual computers in a data centre have more workload than they can handle, while others have less. The workload must be balanced as a result, and efficient virtual machine migration promotes workload balance while minimising server power consumption in the data centre. This ultimately leads to the energy consumption of the data centre being lowered by turning off idle servers. In our study, virtual machine migration and load balancing are performed using the hypervisor KVM. This optimal machine movement and subsequent reduction in host server CPU use results in lower power consumption and improved energy efficiency of data centres in cloud computing.

Keyword:

VM: Virtual machine, KVM- Kernel-based virtual machine, hypervisor, QEMU- Q Emulator

1. INTRODUCTION

Cloud computing, a technical revolution, allows for flexible IT utilisation, particularly networks, storage, servers, services, and applications, in a cost-effective and pay-per-use manner without actually purchasing them [1]. Because it has several advantages for organisations, such as quicker start-up times for new services, lower maintenance and operation costs, higher utilisation through virtualization, and easier disaster recovery, cloud computing is a desired choice [2]. A novel computing model that allows users to rent and release resources (such CPU and storage) via the Internet as needed has been made available thanks to technical improvement [3].

The paper is organized in sections followed by literature review, proposed methodology, experimental setup, results, conclusion, future scope and references.

2. LITERATURE REVIEW ABOUT LOAD BALANCING

Load balancing is one of the main issues with cloud computing. It is a system that evenly distributes the dynamic local workload over every server in the entire cloud, preventing a situation where some servers are heavily loaded while others are idle or doing little work [4]. Due to the trend towards server-side computing and the increasing demand for Internet services, data centres are swiftly becoming into a crucial part of the Internet infrastructure. Data centres are being used more and more frequently by big enterprises, banks, telecom, portal sites, etc. [5]. Data centres will certainly get larger and more complex, creating challenges for deployment, resource management, service dependability, and other difficulties [6]. Virtual machines (VMs) serve as the essential component of a data centre. A virtualized (or virtual) data centre (VDC) is one that uses virtual machines (VMs) as its primary processing components and was built using server virtualization technology [7].

A new paradigm for virtual machine migration was presented by Nagamani H. Shahapurea and P. Jayarekha[8], based on the needs of the customers. A customer asks for a quicker response time. An method known as Virtual Machine Migration Approach Based on Distance and Traffic is created to do this. The technique is based on the data centres' physical locations and traffic patterns. The programme is conducted on a regular basis to monitor network traffic. The distance between the data centres from which the client request must be sent is also checked.

Celia G. Ralha et al. [9] shown how to positively allocate resources in a cloud environment using a multi-agent system. They developed a resource forecasting model based on multivariate linear regression. The suggested approach minimises error rates even though optimisation takes a lengthy time. Cem Mergenci and Ibrahim Korpeoglu devised a multi-allocation method for cloudlets [10]. In this study, TRfit and UCfit, two resource usage measures, were proposed. The effectiveness of resource allocation for load balancing is evaluated using these. By doing this, they can make sure that each virtual machine is using the physical server's resources effectively.

Sobhanayak Srichandan et al.[11] introduced a task-based methodology that combined genetic algorithms and the bacterial foraging algorithm. This essay is divided into two parts. The scheduling method first minimises the Makespan, then in the subsequent phase, it enhances the metrics for convergence and stability parameters for load balancing. According to Hussain A, et al. [12], this recommended technique offers a genetically based way to sustain the load in cloud computing. For population initialization based on their time, the request's priority is taken into consideration. The genetic algorithm consists of three steps: the selection operator, the crossover operator, and the mutation operator. The advantages of this study are successfully utilised in large-scale applications and process utilisation. The shortcomings of this paper do not guarantee finding the optimal solution.

Random resource allocation algorithms are used in this study by Priyanka C.P., et al. [13] to maintain the load in the cloud. Giving the resource a job at random is known as random resource allocation. Before determining how many resources are required to finish the activity, be sure they are all available. The advantages of this effort include preventing resource underutilization and reducing resource waste and electricity use. One of the problems with this paper is that better resource management is required for resource allocation. Reducing the number of resources, like computer servers, needed to support the workloads is the aim of merging separate servers and their workloads. Prices may go down as a result, and peak and average power usage may go down as well.In some data centres, lowering peak power utilisation may be essential if peak power cannot be easily raised [14]. Server consolidation is especially important when user workloads are unpredictable and need to be reviewed frequently. When a user's demand changes, VMs can be scaled and relocated to different physical servers if necessary [15].

3. PROPOSED METHODOLOGY

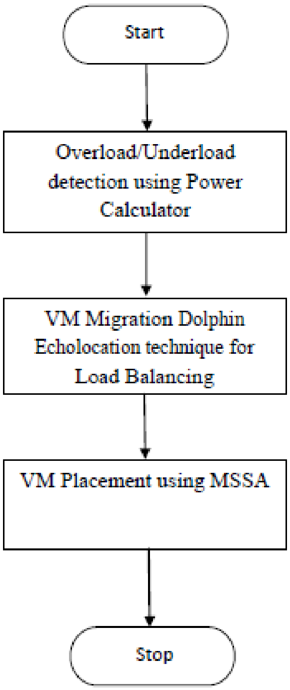
The three steps for load balancing used in the proposed work are as follows: (1) Overload/Underload detection (2) VM Migration (3) VM Placement.

For load balancing and virtual machine migration, the suggested work is utilised to differentiate between the best and worst data centres based on computing time, CPU and network utilisation, as well as hot and cold cloud data centres. Based on the standard data centre layout, user request placement algorithms are utilised to determine the best location for each request on the servers. The power model generates a list of virtual machines that require higher processing power during migration. By balancing the workload, this strategy can help identify virtual machines for load-balancing relocation to other servers.

For the virtual machine migration challenge, a multi-objective Dolphin Echolocation technique is devised. Dolphin echolocation is a method that embraces the type of problem in the immediate vicinity, has a high union rate, and inspires a worthy optimal solution in various circles set by the customer. Acquire a cost-effective system of non-overburdened arrangements that reduces total waste and power consumption. The Tchebycheff algorithm shifts the problem from selecting the most acceptable outcome to selecting the next best choice to the intended outcome.

Finally, an MSSA-based cloud data centre VM placement was demonstrated. The location of the VM was determined by its resource usage and migration expenses, as well as its energy consumption and fitness. SSA is modified to produce VM placement. A new fitness function was also created to select the VM based on migration costs, energy consumption, and resource use.

The following flowchart describes the brief steps of the proposed methodology and algorithms used for load balancing and energy efficiency.



Flowchart: Steps of the proposed methodology used for load balancing and VM Migration and Placement

4. EXPERIMENTAL SETUP

The experiment with a real-time system is carried out using hypervisor [16], an open-source KVM hypervisor that comes standard with Linux and converts the Linux kernel into a virtual machine with all of the characteristics of Linux as well as an array of other features. Some of the highlights include live migration, scheduling, and resource control, as well as improved prioritisation.

A prototype VM management system is created to test the algorithm's performance. The virtualization platform is KVM, while the storage system is NFS. As the host machine, a physical machine is chosen. To manage and schedule the virtual machine, the QEMU-KVM [17] hypervisor and Virtual Machine Manager are installed, and the operating system is Ubuntu 16.04, with an Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70GHz processor and 8 GB of memory. Four client PCs with identical configurations are chosen. The packages libvirt, qemu-kvm, virt-manager, and nfs server have been installed. KVM is installed using the qemu-kvm and qemu-img packages. These programmes offer user-level KVM and disc image management.

4.1 Installation Steps for the Configuration

A. Checking of Virtualization Support on Ubuntu 16.04

Step1: To check if CPU supports hardware virtualization

egrep -c '(vmx|svm)' /proc/cpuinfo

Step 2: To check if a system can use KVM acceleration

sudo kvm-ok

If **‘kvm-ok’** returns an error stating KVM acceleration cannot be used, to **‘cpu-checker’** to solve the problem.

Step 3: To install cpu-checker

sudo apt install cpu-checker

Step 4: Restart the terminal

B. Installation of KVM on Ubuntu 16.04

a) **virt-manager** provides a GUI tool to administrate virtual machines.
b) **libvirt-client** provides a CL tool to administrate virtual environment this tool is called virsh.
c) **libvirt** provides the server and host side libraries for interacting with hypervisors and host systems.

Step 5: To install KVM Packages

1. To update the repositories:

sudo apt update

2. To install essential KVM packages with the following command

sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge utils qemu-img

Step 6: To Authorize the Users

1. Only members of the libvirt and kvm user groups can run virtual machines. To add user to the libvirt group by using the following command.

sudo adduser 'rajesh' libvirt

sudo adduser 'rajesh' kvm

Step 7: verification of installation

1. To confirm the installation by using the virsh command:

virsh list --all

2. systemctl command to check the status of libvirtd:

sudo systemctl status libvirtd

C. Creating a Virtual Machine on Ubuntu 16.04

Step 8: To install **virt-manager**, a GUI tool for creating and managing VMs:

sudo apt install virt-manager

1. To download an ISO containing the OS to install on a VM
2. To start Virt Manager GUI

sudo virt-manager

3. Select the computer icon in the upper-left corner of the window.
4. Select the option to install the virtual machine from an ISO image
5. Navigate to the directory path where the ISO is saved to install it.
6. To proceed, provide RAM and the number of CPUs to the VM
7. To allot hard drive space to the VM and specify the VM's nam
8. Begin the VM instances

Each computer was given a fresh VM for this experiment, which was subsequently assigned to one of four CPU utilisation groups. CPU usage is tracked for each VM and host server. When virtual machines are significantly loaded, the host performs extra work to detect this; if virtual machines are not severely loaded, they are shifted to simplify the environment.

4.2 Choosing and Migrating Virtual Machines

The recommended technique for picking VMs is to use dolphin eco-location algorithms with Chebycheff. With the parameters chosen, this paper estimates the algorithm. When it comes to positioning, each VM's fitness is calculated, and the VMs are then positioned according to the placement guideline. In this case, the migration feature of QEMU-live KVM is used. The libvirt API toolkit known as virsh allows users to use the command to interact with operating system virtualization features.

As seen in Table 1, the virtual machine host H4, which has several virtual machines V_13,V_14,V_15,V_16 running on it, is underloaded, as seen by the power calculator. As shown in Table 2, considering the CPU use load of the host server (H4), its VMs are migrated to host servers H1, H2, H3, and H4 is powered off after a certain number of rounds of applying the migration policy. Disabling a poorly performing server can greatly improve load balancing and energy efficiency.

5. EXPERIMENT RESULTS

Table 1. Host Server and VMs CPU utilization (%) before migration

Host Server	Virtual Machine (VM)	VMs CPU Usage(%) on Respective Core	Host Server CPU Utilization (%)
H1	V_01 V_02 V_03 V_04	45.25 10.09 35.12 8.1	44.78
H2	V_05 V_06 V_07 V_08	30.52 23.08 8.16 48.14	48.17
H3	V_09 V_10 V_11 V_12	21.91 24.20 8.22 10.18	33.90
H4	V_13 V_14 V_15 V_16	15.11 10.98 8.23 2.1	12.43

Table 2. Host Server and CPU utilization (%) after migration

Host Server	Virtual Machine (VM)	VMs CPU Usage(%) on Respective Core	Host Server CPU Utilization (%)
H1	V_01 V_02 V_03 V_04 V_16	44.89 11.23 33.17 9.2 3.1	45.33

H2	V_05 V_06 V_07 V_08 V_15	31.2 23.56 7.85 46.23 8.49	49.67
H3	V_09 V_10 V_11 V_12 V_13 V_14	22.23 25.08 9.35 9.8 14.89 11.87	41.32

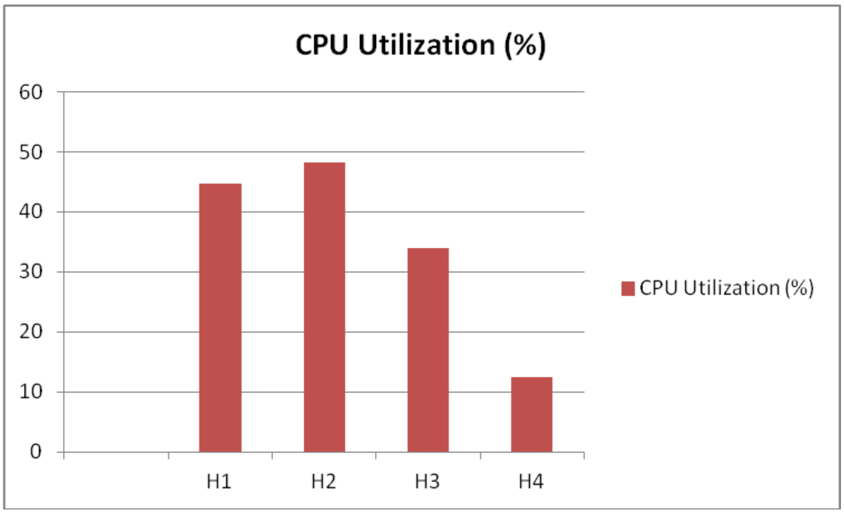


Figure. 1. Host Servers' CPU utilization (%) before migration

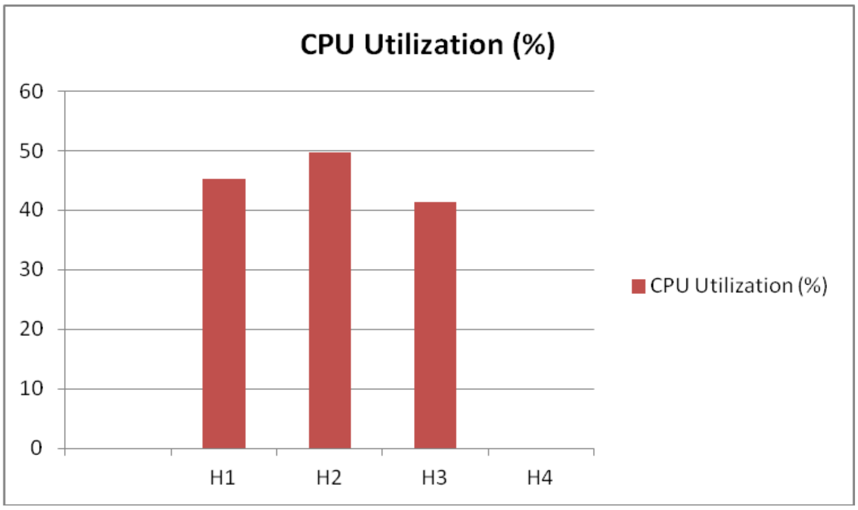


Figure 2. Host Servers' CPU utilization (%) after migration

Figure 1 shows the Host servers' CPU utilization before migration where all host servers H1, H2, H3, H4 are running with moderate or underload conditions. Host server H4 is identified as underloaded by CPU utilization. Figure 2 the Host server H4's VMs are migrated and H4 is put in powered-off mode.

6. CONCLUSION AND FUTURE SCOPE

The proposed work dynamically balances network traffic and lowers energy consumption by turning off idle servers. The network was initially unbalanced. Our proposed methodology discovered the hot and cold data centres using a power calculator, appropriately selected the virtual machine, and used the MSSA (Modified Swarm Salp Algorithm) for VM placement, balancing the workload and designing an energy-efficient data centre. All of the hosts were balanced after several iterations of the algorithm.

Cloud computing is a large field, and load balancing is critical in the case of the Cloud. The work has concentrated on CPU utilisation as a load parameter that is applied to the system, but other metrics such as disc I/O and network demand can also be utilised to balance the load. Further dynamic load balancing can be enhanced, and the Live Migration technique provided in QEMU-KVM can be optimised, reducing migration time and improving performance.

7. REFERENCES:

[1] Vivek Raich, Pradeep Sharma, Shivlal Mewada and Makhan Kumbhka, (2013). Performance Improvement of Software as a Service and Platform as a Service in Cloud Computing Solution. *International Journal of Scientific Research in Computer Science and Engineering,Vol.1, Issue-6, pp.13-16.*

[2] Karthik Kumar and Yung-Hsiang Lu, (2010) Cloud Computing For Mobile Users: Can Offloading Computation Save Energy. *Published by the IEEE Computer Society.*
.
[3] Nabil Sultan, (2010) Cloud computing for education: A new dawn", *International Journal of Information Management, vol. 30, pp. 109–116*

[4] Robert Basmadjian, Hermann De Meer, Ricardo Lent and Giovanni Giuliani, (2012) Cloud computing and its interest in saving energy: the use case of a private cloud. *Journal of Cloud Computing: Advances, Systems and Applications, vol. 1, no. 5.*

[5] Lingjia Tang, Mary Lou Soffa and Jason Mars, (2011) Directly Characterizing Cross Core Interference through Contention Synthesis. ACM 2

[6] Gaurav Dhiman, Giacomo Marchetti and Tajana Rosing (2010) vGreen: A System for Energy-Efficient Management of Virtual Machines, *ACM Transactions on Design Automation of Electronic Systems, Vol. 16, No. 1.*

[7] Xiangzhen Kong, ChuangLin, YixinJiang, WeiYan and XiaowenChu, "Efficient dynamic task scheduling in virtualized datacenters with fuzzy prediction", *Journal of Network and Computer Applications, (2010) (Elsevier).*

[8] Nagamani H Shahapurea, P Jayarekha (2018) Distance and Traffic Based Virtual Machine Migration for Scalability in Cloud Computing, *International Conference on Computational Intelligence and Data Science (ICCIDS 2018) pp.728–737*

[9] C.G. Ralha, A.H.D. Mendes, L.A. Laranjeira et al., (2018) Multiagent system for dynamic resource provisioning in cloud computing platforms, *Future Generation Computer Systems. https://doi.org/10.1016/j.future.2018.09.050*

[10] C. Mergenci, I. Korpeoglu (2019) Generic resource allocation metrics and methods for heterogeneous cloud infrastructures, *Journal of Network and Computer Applications. doi.org/10.1016/j.jnca.2019.102413*

[11] S. Srichandan et al.2018.Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. *Future Computing and Informatics Journal. doi.org/10.1016/j.fcij.2018.03.004.*

[12] Hussain A- Makasarwala and Prasun Hazari. "Using Genetic Algorithm for Load Balancing in Cloud Computing." (2016) *IEEE International Conference on Electronics, Computers and Artificial Intelligence*

[13]Priyanka C.P and Sankari Subbiah (2017) Comparative Analysis on Virtual Machine Assignment Algorithm. *IEEE International Conference on Computing and Communic-ation Technologies.*

[14]George Kousiourisa, Tommaso Cucinottab and Theodora Varvarigoua (2011) The effects of scheduling, workload type and consolidation scenarios on virtual machine performance and their prediction through optimized artificial neural network. *The Journal of Systems and Software, vol. 84, pp. 1270– 1291*

[15] Corentin Dupont, Thomas Schulze, Giovanni Giuliani, Andrey Somov and Fabien Hermenier (2012) An Energy Aware Framework for Virtual Machine Placement in Cloud Federated Data Centres", *ACM*

[16] Hypervisor types: *https://www.vgyan.in/type-1-and-type-2-hypervisor/.*

[17] KVM : *https://www.linux-kvm.org/page/Main_Page.*

Authours:

Dr. Rajesh Patel is having 17+ Years of teaching experience and his research domains are cloud comuting, Sensor Network, AI & ML. Currently working as a Associate Professor in Sankalchand Patel College of Engineering. He is currently guiding 6 Ph.D scholars at University. He has published more than 14 research papers in reputed Internaition journals including scopus indexing and conferences.

Dr Kirit Modi serve as a Dean- Faculty of Computer Science, Vice-Principal, Professor and Head at CE & IT Department, IQAC Coordinator at Sankalchand Patel College of Engineering Sankalchand Patel University. He have more than 22 years of teaching and research experience. He has published more than 40 research papers in the reputed international conferences referred journals.He has serve as a member of editorial board and program committees of International conferences and Journals indexed in Web of Science and SCOPUS.

Mehul S. Patel has 18+ Years of teaching experience and his research areas are Robotics, IoT & AI & ML. Currently working as an Assistant Professor, at the Department of Information Technology at Sankalchand Patel College of Engineering.