International Journal of System and Software Engineering
6 (1), June 2018, 23-29
http://www.publishingindia.com/ijsse

Study of Different Scheduling Techniques to Manage and Conserve Energy in Cloud Data Centers

Shudhanshu Kulshrestha[1*], Tanmay Chaurasia[2], Lakshay Chauhan[2] and Utkarsh Saxena[2]
[1]Assistant Professor, Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.
Email: sudhanshu.kulshrestha@gmail.com
[2]Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India.
*Corresponding Author

Abstract: Cloud computing is a major building block of current technology. The major investment in the area of cloud computing is because the efficient way of delivering required set of software and tools at the doorstep. In this paper we will try to cover different techniques such as DVFS, Round Robin, SJF, VM migration to test the efficiency of cloud data centers in regards to above mentioned techniques. These methods will be evaluated on the basis of three different parameters that are number of Cloudlets, number of VM's i.e. Virtual machine and number of Hosts.

Keywords: Cloud data center, DVFS, Energy efficiency, Process scheduling, SJF.

I. Introduction

The term Cloud computing is derived from a very brief definition of storing information on clouds i.e. the objective of the cloud computing software is to facilitate a better platform where people can rent or buy any kind of software or tools depending on the amount of work. But in this process cloud data centre are prone to a major amount of Carbon Dioxide so our motive is to reduce increase the efficiency of data centers [1]. We will be introduced with the terms like Cloudlet, VM's and Hosts in the Process.

A. Motivation

The motive behind taking up this project was to analyze the areas where cloud data centers are consuming energy and then coming up with solutions to manage the required energy to run the data centre.

Our motive is to find out a scheduling mechanism that can efficiently decrease the amount of energy consumed with respect to time. In the current scenario, we will consider three different objects i.e. Hosts, VMs and Cloudlets on which mechanism will be applied.

B. Research Problem Statement

The Cloud data centre works on the principle of the supplied electrical energy that is actually converted into consumed power. So, the research problem statement is "To find an efficient way to use the resources so that the energy consumed should be minimum and the output be maximum". Our Focus is to find out the best Energy efficient mechanism that can be found to increase the efficiency of the Cloud Data Center.

The main aim of the project is to evaluate the techniques like DVFS and SJF to increase the efficiency and performance of the data centers.

There are three main objectives that are underlined to achieve the required goal.

- Try to find and understand the importance of virtualization.
- Understanding the issues related to Cloud Data Centre management.
- Working of CloudSIM simulator and use it in efficient manner.

C. Scope of Project

Our primary goal was to find out a project that is challenging and requires intense thinking and research. As a result this project based on Cloud computing requires immense knowledge and develop our skill set to further reach. This project is the outcome of several courses taken during the four year Bachelor of Technology process in Computer Science Engineering. One of the course i.e. Introduction to Cloud computing and Web Services provides the basic knowledge that is required to build this project. The skills and processed data that is gained from immense hard work and knowledge of programming languages like Java and C++ were helpful in building up the base of the project. These skills can be enhanced over time to improve the quality of the project [2].

II. Background Study

A. Summary of Other Papers

The cloud computing environment is continuously increasing and giving an edge to data centers. The debate is based on both private and public sector and the main objective is to consider the best of both sectors [3]. We now have access to three different vital utilization process that helps in transmission and user satisfaction, to be clear those three sections are stockpiling as an administration, programming as an administration, and handling as an administration. If we get any new model in future those model will be a mix of these given models to be sure. In cloud programming administrations, control utilization in transport is unimportantly little at low screen invigorate rates. Therefore, cloud administrations are more productive than current midrange PCs for straightforward office assignments. At direct and high screen revive rates, control utilization in transport winds up critical and vitality funds over midrange PCs are reduced [4]. Vitality effectiveness has developed as a standout amongst the most critical plan necessities for present day processing frameworks, for example, server farms and Clouds, as they keep on consuming tremendous measures of electrical power. Aside from high working expenses brought about by registering assets, this prompts critical emanations of carbon dioxide into nature. For instance, at present IT foundations contribute around 2% of aggregate CO2 footprint [5]. Effective power administration in registering frameworks is a notable and widely considered in the past issue. Shrewd administration of assets may prompt critical diminishment of the vitality utilization by a framework, while meeting the execution prerequisites. Unwinding of the execution requirements more often than not brings about additionally diminished vitality utilization.

B. Taxonomy

Virtualization

The process of virtualization is a concept that requires the use of virtual machine that creates a virtual environment for the user. The stages of virtualization requires a setup of cloud environment that can assign various task to the OS working on the different host computer. A virtualized system consumes energy other than the users own system. The most amazing feature of virtualization is that it can show the user library and visitor library simultaneously without any resource lack and fps lack.

Types of Virtualization

- Operating System Virtualization
- Application Server Virtualization
- Network Virtualization
- Hardware Virtualization
- Software or Storage Virtualization

Hypervisor

A hypervisor [8] is a device that permits a certified user to work on a host system present on the operating system framework and keep it running until then the user wants to. The user can share the equipment of the host personal computer such as its operating system and its RAM, to such an extent that it looks like OS is already having its own memory and assets.

A hypervisor is otherwise called a Virtual Machine Supervisor (VMM). There are two kinds of hypervisors: Type 1 and Type 2.

Type 1 hypervisors run straight forwardly on the framework equipment. They are regularly alluded to as a “local” or “uncovered metal” or “installed” hypervisors in merchant writing.

Type 2 hypervisors keep running on a host working framework. At the point when the virtualization development initially started to take off, Type 2 hypervisors were generally prominent. Managers could purchase the product and introduce it on a server they as of now had.

DVFS

Dynamic Voltage and Frequency Scaling (DVFS) is the alteration of energy and speed settings on a registering gadget’s different processors, controller chips and fringe gadgets to improve asset designation for assignments and expand control sparing when those assets are not required.

DVFS enables gadgets to perform required errands with the base measure of required power. The innovation is utilized as a part of all advanced PC equipment to augment control reserve funds, battery life and life span of gadgets while as yet keeping up prepared figure execution accessibility.

VM Migration

Virtual Machine (VM) is an imitating of a specific PC framework. In distributed computing, Virtual machine relocation is a valuable an instrument for relocating Operating System examples over different physical machines. It is used to stack adjusting, blame the administration, low-level framework upkeep and lessen vitality utilization. There are different systems and parameters accessible for VM relocation.

FCFS

In the “First come first serve” algorithm, as the name proposes, the procedure which arrives first, gets executed first, or we can state that the procedure which asks for the CPU first, gets the CPU assigned first.

First Come First Serve, is much the same as FIFO (First in First out) Queue information structure, where the information component which is added to the line first, is the person who leaves the line first.

SJF

In the “First come first serve” algorithm, as the name proposes, the procedure which arrives first, gets executed first, or we can

state that the procedure which asks for the CPU first, gets the CPU assigned first.

First Come First Serve, is much the same as FIFO (First in First out) Queue information structure, where the information component which is added to the line first, is the person who leaves the line first.

Maxmin

This is an efficient algorithm that allocate a particular task to a given host which ensures the maximum utilization of resource to the minimum value. The tasks are first queued up in the task queue and then they are selected from there.

Hosts

A network host is a PC or other gadget associated with a PC network. A network host may offer data assets, administrations, and applications to clients or different hubs on the network. A network host is a network hub that is allotted a network address.

Cloudlets

A cloudlet is a versatility upgraded little scale cloud server farm that is situated at the edge of the Internet. The fundamental motivation behind the cloudlet is supporting asset escalated and intuitive versatile applications by giving intense figuring assets to cell phones with bring down idleness.

VM's

A virtual machine can be defined as a image or a replica of a computer which can handle all arithmetical and logical process or a Virtual machine is often referred as a framework that is required to work on the operating system which is helpful in maintaining the programming environment with the help of committed equipments. The motive is to make such an aura that end client can't distinguish between personal computer and a virtual machine.

III. Cloudsim

IT organizations who will offer a few administrations in the Cloud can utilize a recreation based way to deal with play out some benchmarking explores different avenues regarding the administrations to be offered in reliable, versatile, repeatable, and controllable situations previously genuine sending in the Cloud. Accordingly, they can test their administrations in a controlled domain free of cost, and through various emphases, with less exertion and time. Additionally, by utilizing recreation, they can do distinctive investigations and situations to distinguish the execution bottlenecks of assets and create provisioning systems before genuine sending in business Clouds.

In this way, CloudSim has been produced to satisfy these necessities by reproducing and displaying extensible Clouds.

A. Architecture

The toolkit Cloudsim is characterized as "another, summed up, and extensible recreation system that permits consistent demonstrating, re-enactment, and experimentation of developing Cloud Computing foundation and application administrations". Also the cloudsim toolkit is build up with a lot of layered architecture such as a simple layer of Simjava to the more complex layer of User code.

<table>
<tr><td colspan="6">User Code</td></tr>
<tr><td>Simulation Specification</td><td colspan="3">Cloud Scenario</td><td colspan="2">User Requirement</td></tr>
<tr><td>Scheduling Policy</td><td colspan="5">User or data centre broker</td></tr>
<tr><td colspan="6">CloudSim</td></tr>
<tr><td>User Interface Structure</td><td colspan="3">Cloudlet</td><td colspan="2">Virtual machine</td></tr>
<tr><td>VM Service</td><td colspan="3">Cloudlet Execution</td><td colspan="2">Management</td></tr>
<tr><td>Cloud Service</td><td>VM</td><td>CPU</td><td>Memory</td><td>Storage</td><td>Freq.</td></tr>
<tr><td>Cloud Resource</td><td colspan="3">Event Handling</td><td colspan="2">Cloud Conductor</td></tr>
<tr><td>Network</td><td colspan="5">Network Topology</td></tr>
</table>

Fig. 1: Initial Layered CloudSim Structure

B. Usability

With a specific end goal to utilize the CloudSim toolbox, clients need an essential foundation in Java programming dialect since it is composed in Java. Likewise, it expects clients to keep in touch with some code to utilize the parts from its library to reproduce the coveted situations. In this way, it is not only about giving the parameters to the system and executing the given program, and gathering the outcomes, however, it likewise requires a profound comprehension of how the program functions. Also learning more about the IDE's like Netbeans and Eclipse will give the user an edge and will be helpful in making the Cloudsim toolkit easier to use and give the result.

C. Limitation

CloudSim is an effective instrument for displaying and recreating Cloud registering, however, it has a few impediments. Right off the bat, it's anything but a prepared to-utilize instrument that that would simply require setting parameters as it were. All things considered, it requires thinking of some Java code to utilize its library, as talked about prior. Additionally, the abilities of CloudSim are once in a while constrained and require a few augmentations. Utilizing the library keeping in mind the end goal to ease demonstrating by just concentrating on the many-sided quality of the re-enacted situation, without investing much exertion and energy in the dialect in which the test system is deciphered.

IV. Configuration Details

The philosophy will be founded on utilizing a reproduction to assess the components that can be utilized to enhance the eco-proficiency of data centre. Along these lines, the advancement approach of this venture will begin with breaking down and outlining a situation of the recreated try, running the situation, checking the outcomes, lastly, assessing the outcomes.

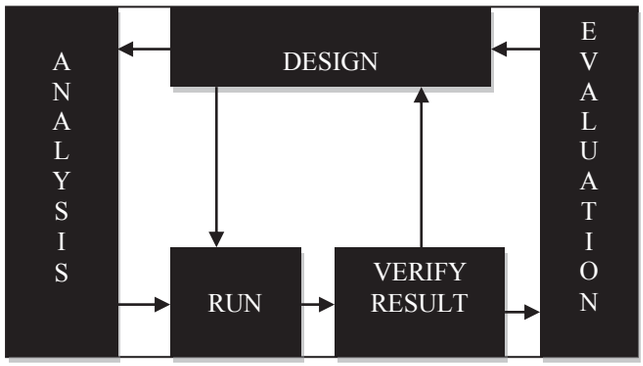

Fig. 2: Interactive Approach for Experiment and Development

A. Hardware and Software Requirement

- First of all we need to download the Cloudsim and the latest version of the JDK. (Note: It is advisable to download the appropriate version i.e. x86 or x64 based on your system specification).
- Cloudsim requires the working JRE so install the JDK.
- Install a Java environment platform such as Netbeans or Eclipse.
- Now Install Cloudsim. Unpack the downloaded 'Cloudsim-3.0.3.zip'.

B. Setting Up Environment

- First set the path variable which contains files like 'javac' and 'java' which will be used by Cloudsim while compiling.
- Then set the CLASSPATH variable which contains the location of class file.
- My Computer --> Advanced system settings --> Advance tab --> Environment Variable --> under the System Variable --> New -- > CLASSPATH.

C. Testing the Setup

- Compile a cloudsim program i.e. 'javac filename.java'.
- To run the compiled program type 'java org.cloudbus.cloudsim.filename'.
- To save the output type 'java org.cloudbus.cloudsim.filename > output.txt'.

D. Working on Simulation

The Scenario will include the testing of experiments with and without the deployment of energy conservation mechanism. The first part will include working on scheduling techniques such as FCFS and SJF and plotting the comparison graph between all the techniques. The Second phase will include working on DVSF and VM migration Mechanisms and finding out the result based on the number of cloudlets and hosts.

Energy Consumption = (Finish Time - Start Time)*(Host Number*Max Power)*(VM number*Bandwidth)*(Cloudlet Number*Length)/(MIPS Hosts*MIPS VMs*MIPS Cloudlets)

Fig. 3: Implemented Formula for Energy Conservation

Table I: Typical Characteristic of Hosts, Cloudlets, VM's

Hosts	
Max Power	350 watt
Static Power	0.7(70%)
MIPS	{1100,2200,3300}
RAM	10000 MB
Storage	1000000 MB
Bandwidth	10000 Mbps
VM's	
MIPS rating	{350,700,1250,1700}
No. of CPU	1
RAM	256 MB
Bandwidth	2500 Mbps
Image Size	2750 Mbps
Cloudlets	
Length	150000
No. of CPU	1
File size	300
Output size	300

V. Results

A. FCFS Scheduling

OUTPUT:

Cloudlet ID	Status	Data Centre ID	VM's ID	Time	Start Time	Finish Time
0	Success	2	0	67.27	0.1	67.37
1	Success	2	1	44.49	0.1	44.50
2	Success	2	2	34.33	0.1	34.35
3	Success	2	3	8.89	0.1	8.90
4	Success	2	0	39.99	0.1	40.00

Cloudlet ID	Status	Data Centre ID	VM's ID	Time	Start Time	Finish Time
5	Success	2	1	13.5	0.1	13.6
6	Success	2	2	16.66	0.1	16.76
7	Success	2	3	12.22	0.1	12.32

B. SJF Scheduling

OUTPUT:

Cloudlet ID	Status	Data Centre ID	VM's ID	Time	Start Time	Finish Time
3	Success	2	3	8.89	0.1	8.99
7	Success	2	3	12.22	0.1	12.32
5	Success	2	1	13.5	0.1	13.6
6	Success	2	2	16.66	0.1	16.76
9	Success	2	1	20.5	0.1	20.6
2	Success	2	2	34.33	0.1	34.43
8	Success	2	0	38.17	0.1	38.27
4	Success	2	0	39.99	0.1	40.00

C. Roundrobin Scheduling

OUTPUT:

Cloudlet ID	Status	Data Centre ID	VM's ID	Time	Start Time	Finish Time
3	Success	2	0	8.89	0.1	8.99
7	Success	2	1	12.2	0.1	12.32
5	Success	2	2	13.5	0.1	13.6
6	Success	2	3	16.6	0.1	16.76
9	Success	2	0	39.9	0.1	40.00
2	Success	2	1	13.5	0.1	13.6
8	Success	2	2	16.6	0.1	16.76
4	Success	2	3	12.2	0.1	12.32

D. Maxmin Mechanism

OUTPUT:

Cloudlet ID	Status	Data Centre ID	VM's ID	Time	Start Time	Finish Time
3	Success	2	3	8.89	0.1	8.99
7	Success	2	3	12.22	0.1	12.32
5	Success	2	1	13.5	0.1	13.6
6	Success	2	2	16.66	0.1	16.76
9	Success	2	1	20.5	0.1	20.6
2	Success	2	2	34.33	0.1	34.43
8	Success	2	0	38.17	0.1	38.27
4	Success	2	0	39.99	0.1	40.00

Min-Min finished!

BUILD SUCCESSFUL (total time: 0 seconds)

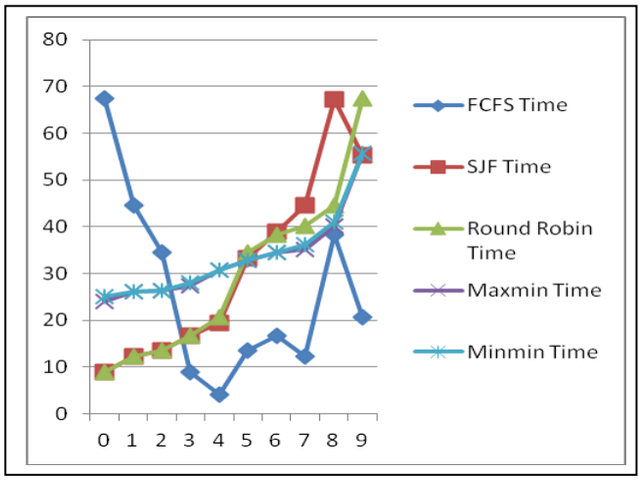

Fig. 4: Comparison Graph for all Scheduling Process

E. DVSF Mechanism

Increasing the number of Hosts, VMs and cloudlets.

TABLE II: EXPERIMENT WITH DEPLOYMENT OF DVSF MECHANISM

Experiment	Parameter Setting	Value of Energy Consumption
1	No. of Host = 1 No. of VM = 2 No. of Cloudlet = 2	0.245714
2	No. of Host = 2 No. of VM = 4 No. of Cloudlet = 4	0.50
3	No. of Host = 4 No. of VM = 8 No. of Cloudlet = 8	1.005714

TABLE III: EXPERIMENT WITHOUT DEPLOYMENT OF DVSF MECHANISM

Experiment	Parameter Setting	Value of Energy Consumption
1	No. of Host = 1 No. of VM = 2 No. of Cloudlet = 2	0.87
2	No. of Host = 2 No. of VM = 4 No. of Cloudlet = 4	1.788571
3	No. of Host = 4 No. of VM = 8 No. of Cloudlet = 8	3.57714

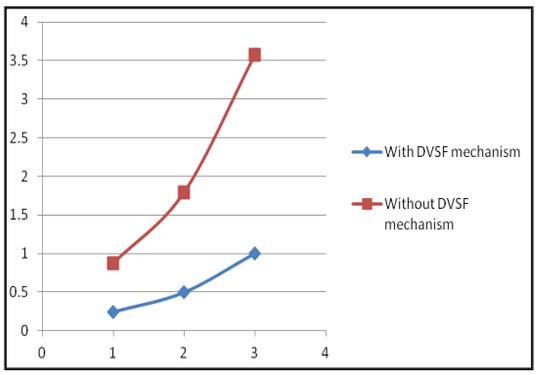

Fig. 5: Comparison Graph

Finally, it is visible through the experiment that applying scheduling mechanism can lower the value of consumed electricity and enhance the performance of the data centre. The data centre deployed with the scheduling mechanism such as DVSF, Roundrobin, FCFC and VM migration takes less energy than the data centre which employ no scheduling techniques , that shows that how efficiently the scheduling mechanisms has been constructed to dynamically control the input voltage and clock frequency of a CPU to decrease the power consumption.

Table IV: Benchmarking of Energy Consumption Between the Deployment of DVSF Mechanism and without Any Mechanism in a Data Centre

	Variation of energy consumption for each subsequent experiments with Scheduling Mechanism	Variation of energy consumption for each subsequent experiments with NPA	Average energy saving
Doubling the number of Hosts, VMs and cloudlets	Double	Double	About 70%
Doubling the number of Hosts	Stable	Double	Varies (Dramatically increase in energy saving with increase in number of hosts)
Doubling the number of VMs	Almost Stable	Stable	About 70% in experiment 1 and 16% in subsequent experiment
Doubling the number of cloudlets	Stable	Stable	About 70%

VI. Conclusion

As the amount of utilization of Cloud data centers are continuously increasing, the vitality of figuring assets are on the brim of expanding, for example, server farms, which support the Clouds, will increment also. This would build the weight of the operational expenses and the natural effect if there is no influence over the vitality utilization. In this way, controlling and decreasing the vitality utilization in server farms would not just lower the operational costs, which consequently would build the benefits, yet additionally, bring down the ecological effect regarding CO2 emanations.

Along these lines, this undertaking has researched and assessed system, in particular, DVFS, SJF, FCFS, Roundrobin instrument, which can be conveyed in server farms to control the vitality utilization and enhance their productivity.

Limitations

The experimental mechanism applied on this project is limited to the measuring of the metric value of energy consumption only. However, if another experimental metric such as SLA violation was included in the calculation process of our project, it would have increased the quality of our project. But due to lack of proper instruction and knowledge about CloudSim toolkit before starting the project. It was very tough to include another experimental criteria such as SLA violation, mostly due to the restricted time scale for project. Therefore, the evaluation of FCFS, DVSF, SJF, Roundrobin and VM migration can be enhanced by considering another mechanism side by side, such as SLA violation.

At the end, some of the simulated results could have similar results at different points due to the fact that these results were not based on specific need and real environment but implemented on the basis of exemplary requirement and uses a typical random value settings.

Future Work

In future, we might want to utilize another booking instrument i.e. VM relocation and plot chart to discover which of the two DVFS or VM movement component is better in long run.

Notwithstanding beating confinements specified above, coordinate tests including the situations could be set up in a research facility to acquire more precise outcomes for estimating the execution of vitality cognizant systems, for example, DVFS and VM assignment, in a genuine domain. Be that as it may, the improvement approach would be diverse to that utilized as a part of this venture.

This task has concentrated on assessing the systems that could be utilized as a part of a server farm as a method for enhancing the eco-proficiency of Cloud Computing as far as the foundation layer. Notwithstanding, the effectiveness of Cloud data centers for various layers of operating systems such as Network and Transport layer. For instance, of a person runs the virtual clone of a layer on his device using a virtual environment provided

by cloud data centers, it would contribute towards a better cause and improve the usability of Cloud computing.

References

[1] Ibrahim Alzamil, MSc Computing and Management Session (2011/2012).

[2] K. Arnold, J. Gosling, and D. Holmes, *The Java Programming Language*, Addison Wesley Professional, 2005.

[3] P. Mell, and T. Grance, The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law, pp. 107-347, 2002.

[4] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7-18, 2010.

[5] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," In *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149-167, January 2011.

[6] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomay, "Energy-efficient data replication in cloud computing datacenters," *2013 IEEE Globecom Workshops (GC Wkshps),* pp. 445-450, 2013.

[7] 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*, vol. 82, pp. 47-111, 2011.

[8] G. Banga, P. Druschel, and J. C. Mogul, "Resource containers: A new facility for resource management in server systems," In *Proc. 3rd OSDI*, pp. 45-58, New Orleans, LA, February 1999.