Journal of Scientific & Industrial Research
Vol. 79, June 2020, pp. 526-530

Efficient Utilization of Virtual Instances by Suspend Resume Strategy in Cloud Data Center

A Nirmal Kumar*

Department of Computer Science and Engineering, CMR Institute of Technology, Hyderabad, Telangana, India

Received 1 June 2019; revised 17 February 2020; accepted 3 April 2020

The effective utilization of virtual instances by suspend resume policy in virtualized data center is analyzed in this paper. Cloud computing is a term that describes the means of delivering all information technology from computing power to computing infrastructure, applications, business process and personal collaboration to end users as a service wherever and whenever they need it. Here Infrastructure as a Service is used for open source cloud implementation. Open Stack provides architecture for cloud to build the virtual instances. Thus Virtual Machine allocates a single job, by dividing them to the grid systems. Suspend resume policy is used to provide the jobs to the virtual instances based on the usage. It helps in examining the weight flanked by the virtual instances as job arrived.

Keywords: Cloud, Virtual data centers, Openstack, Suspend resume policy

Introduction

The cloud in distributed computing is the arrangement of equipment, programming, systems, stockpiling, administrations and interfaces that join to convey parts of processing as assistance shared assets programming and data that are given to Personal Computers and different gadgets on request. It permits individuals to do things they like to do on a Personal Computer without the requirement for them to purchase and construct an Information Technology foundation on to comprehend the basic innovation.

There is a great need for it to help addressing, business challenges. Cloud computing can help in doing business more with less expenditure, providing high quality service, reducing risk, and ensuring breakthrough agility. The characteristics of the cloud are on-demand computing and ubiquitous computing etc. The services in the cloud are of 3 types. They are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Service as a Service (SaaS). Grid computing is a group of computer from different locations to arrive at a universal goal. The network can be determined as a disseminated arrangement.

The open stack cloud architecture will build the Virtual Machine (VM). Virtual Instance (VI) is shaped in OpenStack by means of two methods. Virtual Instance is shaped openly in OpenStack. Instance is created in OpenStack using dashboard or using j cloud Application Program Interface (API). If a solitary work is allocated to single virtual machine, power utilization will be very high and the response time will be more. Thus the work is alienated into manifold jobs. These jobs are allocated to the Grid of virtual instance. The job therefore reduces the response time and suspend resume policy is implemented to provide efficient energy consumption which will allocate the virtual machine based on the command. It helps in harmonizing the weight between the instances as the work arrives. There are totally 4 modules involved in it. They are OpenStack configuration, build and configure instance, launch an instances, and power management in grid computing.

Existing system

Cloud Computing is a term that describes the means of delivering any and all information technology from computing power to compute infrastructure, applications, business process and personal collaboration to end users as service whenever and wherever they need it. Here IaaS is used and the energy consumption is optimized by using three power saving policies to reduce the idle power of the server. Here N policy is used, in that when server machine will be turned on, only when the number of jobs in the queue is more or same as that of the specified N threshold. The three policies used are Integrated Server Nodes (ISN) policy, System Integration (SI) policy and Server Nodes (SN) policy.

*Author for Correspondence
E-mail: nirmalkumar@cmritonline.ac.in

In the Integrated Server Nodes, policy the server will enter into the busy mode when the job arrives and it will end its busy mode only when it completes all of its jobs. The server will be in the sleep mode until the number of jobs in the queue id is less than that of the specified N threshold value. In the Server Nodes, the server will enter into the sleep mode only when there is no job in the specified machine, whereas in the System Integration policy, the server will be in the sleep mode for only a certain amount of time when the sleeping time of the server ends, it will enter into the idle mode. Thus, the consumption of idle power is efficiently reduced at the same this, thus more policies are used and in this the beginning cost will be high.[1]

Here cloud as a service oriented platform dynamic allocation of resources is proposed by means of virtualization technology. In this technology, the resources are allocated dynamically to the system according to the need of the user. In order to allocate resources dynamically to the system and to determine the utilization of uneven multiple resources skewness is used. By using this skewness, the load between the virtual machine is balanced. Thus the overloading between the virtual machines is prevented by effective load balancing between the machines and at the same time physical machines should be capable enough to handle all the resource needs of the virtual machine, else overloading of the physical machine takes place and the virtual machine performance is decreased.[2]

For the past few years, consumption of energy has become the most difficult part in computing environment especially their data centres. Here the energy efficiency is improved in the web servers by determining the daily requests patterns sent to the web servers which indicate the resource provisioning in offline and will be more suitable for management of resources. Here two load distribution algorithms are used. They are relative load distribution, servers are utilized equally and the adaptive load distribution, the nodes are selected based on the power level. Mostly these two algorithms are used to view the usage of the Central Processing Unit (CPU) and level of power consumption in the web servers.[3] Thus, at the end, shutting down of the idle nodes will save the energy. It is however difficult to make out this offline methodology without the daily request patterns. Thus in future it should be enhanced in such a way that the workload predictions should be based on online resource management.

The cloud computing has generally reshaped the cloud computing. Though the cloud provides more advanced features, still it is lagging in its operational cost because it is very high for both the public and the private clouds. Green computing is also very important with minimum resource and more demand. A framework is provided that ensures enhancements in the cloud architecture.[4] By using less number of virtual machine, power aware scheduling techniques, proper resource management, and the cloud data center can be improved with minimum overhead. Thus, future enhancements should be done in such a way that all the fields that the proper resource management, scheduling techniques should be enhanced.

Data centers generally consume most amount of energy which is determined in terms of distribution of power and cooling of the systems. By dynamically adjusting the active machines, the resources can be efficiently used according to the system needs. Generally the data centers will have a wide variety of computer systems with varying energy consumption characteristics. Then, works performed by the system will be generally based on the priority, resource needed and the objective. Failure in any one of these characteristics will lead to a problem, thus here the heterogeneity aware resource management is presented for providing a dynamic work in the clouds. K means clustering algorithm is used for allocating the workload based on the similar characteristics in terms of requirements and resource. Here a novel technique is presented for adjusting the machines dynamically to minimize the power.[5]

Data centers consume more resources in the cloud computing epoch, which can be minimized by erasing the number of turned-on servers. A power-conscious scheduling technique for virtual data center resources uses this algorithm that periodically consolidates servers and changes the virtual machine placements in consolidation rounds to satisfy a Service Level Agreement (SLA). A power-aware datacenter simulator is constructed and validated using this methodology. Using this, Scheduling Algorithm simulator performance is analyzed. This strategy is more power efficient than the one based on events. Further simulation results show that in general, workload-based weight adjustment by virtual machine can allow power-aware schedulers to perform more efficiently under the SLA. This strategy increases the data center control budget by 35% for web-like

unintended workloads, and the data center progress budget by 22.7% for workloads with secure reserve criteria like ScaLAPACK.[6]

Cloud task consolidation has become an important strategy in streamlining resource usage and in effect improving energy efficiency. Resource utilization relates directly to energy use, modeled their relationship and established two energy-conscious task heuristics consolidation.[7] Monitoring and controlling the behavior of Energy Management System (EMS) by fulfilling its purpose at the same time balancing certain constraints, including cost optimization, system reliability and environmental consideration.[8] Examples of systems are building energy management, micro grid energy management and datacenter energy management. It shows promise in the improvement of energy savings across many domains. EMS is a proof of implementation in a wide variety of uses.

Electricity expenditures at Internet Data Centers (IDCs) are becoming an important category of operating costs. The overall energy cost minimization of IDCs in the smart grid environment is addressed here. The price maker in the electricity market should be high when the energy consumption is high. IDC will impact electricity price.[9,10] The price varies with the load and the customers according to many observations. Then the cost of energy minimization issue is subjected to end-to-end delay constraints and energy optimization. Performance is evaluated to make the solution method more effective to turn the problem into a quadratic programming. The result shows that the total electricity cost of IDCs is minimized by handling the interaction between IDCs and smart grid.

Cloud computing is used as a service over the Internet to prove dramatically scalable and virtualized resources. Datacenters involve the realization of cloud computing in virtualization technology. As the server's rapid growth, quantity and scale in the datacenter, it is a huge challenge that is directly related to the number of servers hosted and their workloads. Designing and implementing the energy-efficient technologies for the 10 data center is very critical. Possible consolidation of server overhead performance investigates the consolidation strategies and explores the live migration process. These findings demonstrate that both systems are used with lower overhead efficiency to meet the energy-saving targets.

Research methodology of implementation

Proposed framework executes improved energy-effective green control (EGC) calculation which utilizes Suspend Resume arrangement for improving vitality effectiveness in lattice registering. Proposed framework utilizes open source cloud execution and OpenStack gives cloud design which is the best to fabricate the virtual instance. Virtual Instance is made in OpenStack utilizing two different ways:

- Virtual Instance is made legitimately in OpenStack Dashboard
- Using j clouds Application Program Interface, occurrence is made in OpenStack

In the event that a solitary occupation is dispensed to one virtual instance at any given moment, control utilization will be high and reaction time will be postponed. So as to give vitality productivity and low power utilization in a cloud, work is isolated into different little undertakings. These assignments are distributed to a framework of virtual instance. Matrix of virtual instance performs employment to lessen reaction time. The architecture of virtual machine migration is represented in Fig.1.

Suspend Resume approach is executed to give an effective green control which distributes virtual instance dependent on the utilization and can powerfully summon the virtual instance on demand. It additionally helps in adjusting the heap between the virtual instances as the activity arrives. The modules incorporate OpenStack arrangement, manufacture and design occurrence, dispatch a case and power the executives in framework registering.

A. Openstack Cloud Setup

In this module, router configuration will be completed. The following setting is done: switch IP

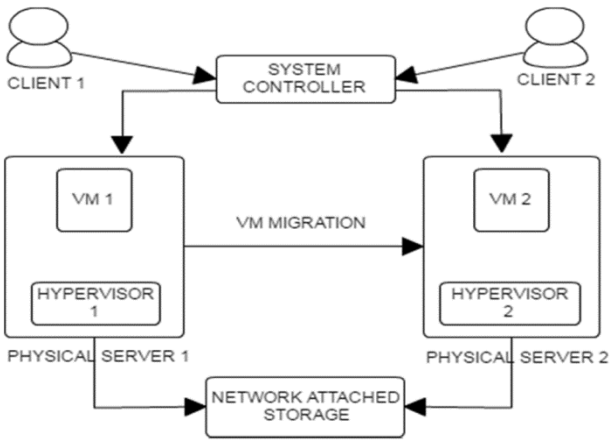

Fig. 1 — Architecture of Virtual Machine migration

address and Wireless Local Area Networks (WLAN) settings. After Router setup, virtualBox is introduced. Circle space will be distributed to virtualBox. Presently virtual apparatuses are imported in VirtualBox. Next processor and connector settings should be set. Confirmation step will be done in openStack. To begin server, the accompanying order to be utilized. /Stack. Sh. Server begins and it gives OpenStack IP and validation subtleties in terminal. To stop server, the accompanying order is to be utilized. /Unstuck. Sh.

B. Build and Configure Instance

In this module, picture and Instance will be made in openStack dashboard. To open dashboard, open program and type openStack IP. The dashboard gets opened. Give username and secret phrase and the page get diverted to openStack administrator page. On the left half of the dashboard, under administrator tab, click pictures connect. In picture connect, click make picture catch. A discourse box gets opened. Give name, depiction, select record in picture source and transfer a picture from the framework and select the arrangement as QCOW2.

At that point click open checkbox. Presently click make picture catch. The picture gets transferred to openStack. To dispatch an occasion, the accompanying strides to be pursued: at the left half of the dashboard, under venture tab, click cases connect. Snap dispatch an occurrence at the upper right of occasion page. Select accessibility zone as a nova, give occurrence name, season, boot from picture as case boot source and pick picture. Presently click dispatch catch. A virtual instance gets propelled in openstack dashboard.

C. Grid Computing and Application Development

In this module, example is propelled utilizing jclouds API. To dispatch an occasion in openStack utilizing jclouds API, give confirmation and openstack-nova supplier to jclouds. At that point give RAM, picture name and example name. Utilizing nodeMetaData, occasion will be propelled in openStack dashboard. A different IP will be made for every virtual instance.

To develop a Grid of virtual instance, administrator needs to give the absolute number of virtual instance to run, beginning virtual instance to be in running state and RAM determination. A network of examples will be made one by one for the absolute number of cases. The base number of occasions to be run is held and other virtual instance is put to rest state. The client can exchange a web application to virtual instance. Presently client can send their web applications in virtual instance. Burden adjusting and Suspend Resume approach isn't actualized on virtual instance in Grid Computing which may prompt high over-burden and at last crashes the server examples. The power utilization and reaction time will likewise be high.

D. Power Management Grid Computing

Administrator will distribute occupations to virtual instance. This activity will put on First Come first Serve (FCFS) line and can be served in framework figuring condition. Occupation is part into little undertakings and designated to framework of running virtual instance. The memory for every virtual instance is observed ceaselessly to counteract over-burdening. Edge esteem will be checked with memory use and if any virtual instance surpasses, it will be accounted for to cloud administrator. Burden adjusting is accomplished by setting off the Suspend Resume Policy to stack another virtual instance which is in the rest state. This guarantees uniform circulation load among all the virtual instances that helps in avoiding high memory use which will definitely impact control utilization.

Result analysis

The evaluation of performance is carried out by number of experiments, and those are listed below. The installation of Kernel Virtual Machine (KVM) is depicted in Fig. 2. We will use a 64-bit kernel. A 2 GB RAM confinement for a given virtual machine will occur on a 32-bit kernel establishment. The order above causes the client to check whether or not the given part of the virtual machine bolsters 64-bit. On the off chance of 0 appearing as output, it means the

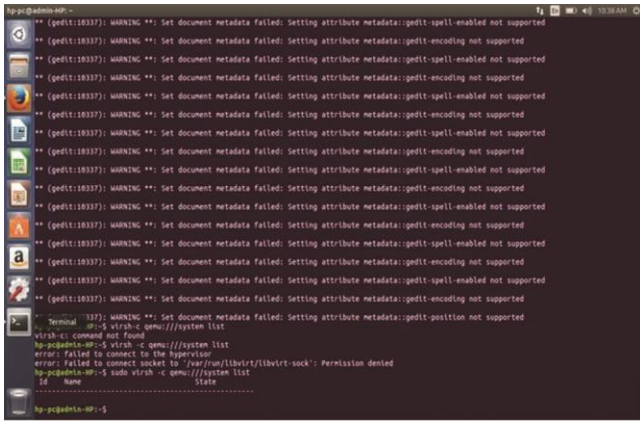

Fig. 2 — Execution of Kernel Virtual Machine

CPU isn't 64-bit. If it is 1 or higher then it is a kernel of 64-bit. The following command uname -m is used to check whether the running kernel is less than or equivalent to 64-bit.

If the output is x86 64 it shows the running kernel to be a 64-bit kernel at that point. In case the client sees other than that the client runs a 32-bit kernel at that point. The 32-bit kernel should not be utilized in light of the fact that it won't bolster the virtual machine. The five packages that are to be introduced are

- qemu-kvm
- libvirt-bin
- bridge-utils
- virt-manager
- qemu-system

The first module clarifies how kernel virtual machine (KVM) is established.[1] The initial step is to check if the CPU supports virtualization of the equipment.[3] The command utilized to check the equipment support is egrep-c '(vmx|svm)'/ proc/cpuinfo. The subsequent stage is to check whether the processor is 64-bit kernel. The command utilized for this rationale is egrep-c 'lm'/proc/cpuinfo.

The processor is not 64-bit kernel on the off chance that the output gets is 0. The processor is a 64-bit kernel on the off chance that the result obtained is 1 or more than 1, then. The next step is to introduce 5 bundles of vital aid to the KVM. It is not necessary to run a 64-bit kernel on the portion to make more than 2 GB of RAM available for the virtual machine which the client must render.

The qemu-kvm is used in the virtualization to run the information and yield. The device used to associate to the virtualization product is the libvirt. Libvirt is useful for the qemu-kvm run. It underpins numerous wide scope of hypervisor. As the VM supervisor the virt-manager is extended. Virtual machine supervisor is the Driving VM user interface. The qemu-kvm is linked with the libvirt-bin through bridge utils. In the design paper, the progressions about the development of virtual machine, holding alive break must be updated. You can open the design document by using the sudo vim /etc / libvirt / libvirtd.conf command. Once all 5 packages have been set up, make the necessary changes in the setup document and then test the establishment using the $virsh -c qemu:/system list accompanying order.

Conclusion and future enhancement

The energy competence is achieved in openstack cloud by means of enhanced EGC algorithm by using grid computing to allocate the jobs to the grid of virtual instance by subdividing the single job into multiple jobs by means of divide and conquer algorithm. The jobs are selected based on the job which comes first, determined by using the FCFS algorithm. By using the SR policy, the virtual instance from the sleep mode is waken to busy mode when any one of the virtual instance has reached its threshold level and in need of another virtual instance. Thus here the energy is efficiently used and since the job is divided by grid computing the response time will be less.

References

1 Ahmad M, Hamid S & Madani, Virtual machine migration in cloud data centers: a review, taxonomy and open research issues, *J Supercomput*, **71** (2015) 2473–2515.

2 Haifeng J, Yanjing S, Renke & Hongli Xu, Fuzzy-Logic-Based Energy Optimized Routing for Wireless Sensor Networks, *Int J Distrib Sens Netw*, **1** (2013) 1–8.

3 Richa S, Vasudha V & Umang , EEFCMDE: energy-efficient clustering based on fuzzy C means and differential evolution algorithm in WSNs, *IET Commun*, **13** (2019) 996–1007.

4 Kumar A N, Kumar V, Efficient performance upsurge in live migration with downtown in the migration time and downtime, *Clust Comput*, **1** (2018) 455–467.

5 Ravikiran N & Dethe, Improvements in Routing Algorithms to enhance Lifetime of Wireless Sensor Networks, *Int J Comp Net Commn*, **10** (2018) 1–10.

6 Ola A, Laila N & Etimad F, Fuzzy Clustering for Next Generation Wireless Sensor Networks, *Int J Adv Rsrchs in Comp Sci Elect Engg*, **8** (2019) 1–8.

7 Parwekar P & Rodda S, Location of Sensors by Base Station in Wireless Sensor Networks, *J Sci Ind Res*, **77(2)** (2018) 83–86.

8 Saravanan P & Kalpana P, A Novel Approach to Attack Smartcards Using Machine Learning Method, *J Sci Ind Res*, **76(2)** (2017) 95–99.

9 Singh M M & Basumatary H, MERAM-R: Multi- Clustered Energy Efficient Routing Algorithm with Randomly Moving Sink Node, *J Sci Ind Res*, **77(1)** (2018) 15–17.

10 Hassan K L, Mandal J K & Mondal S, Enhanced Trust-based Intrusion Detection System in MANET**,** *J Sci Ind Res*, **78(8)** (2019) 509–512.