International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 1, Issue 1, November 2014

VM Provisioning Through Autonomic Broker

[1]M. SrinidhiBhandary,[2]Mr.ShreenathAcharya
[1]Department of Computer Science and Engineering, [2]Department of Information Science and Engineering,
[1][2]St. Joseph Engineering College,Vamanjoor, Mangalore, Karnataka
[1]srinidhibhandary90@gmail.com, [2]shree.katapady@gmail.com

***Abstract*- The movement of services to the internet has resulted in increased demand for the cloud computing. To meet this growing demand, large scale virtualization datacenters need to be established which consume a very large amount of energy in order to provide good quality of service and be reliable. The large energy consumption leads to proportionally large operational cost which affects both the service provider and the service users. In addition to this datacenters emit a large amount of carbon dioxide which would increase global warming. A lot of times the virtual machines (VM) are not utilized completely by simply creating a new VM for newer requests. The proposed mechanism maximizes utilization by checking if the VM can run the new request along with the tasks that are already running with considerable reduction in power consumption, thereby leading to prevention of an additional cause for global warming.**

***Keywords*— Virtual machine, Soft real time provisioning, Hard real time provision, VM Sprawl, Virtualization, power consumption, cost.**

I. INTRODUCTION

Cloud computing has emerged from Grid computing and distributed computing so it inherits most of its features from both the technology. A cloud encapsulates in it all necessary resources such as datacenters, repositories, cooling systems and uses virtualization. The datacenters may exist virtually or physically and tends to consume more power. A widely adopted method of addressing the resource consolidation problem is using a computing system where energy consumed by the computing resources are proportional to the workload of the application. The proposed approach is implemented through the widely adopted Dynamic Voltage and Frequency Scaling (DVFS) technique towards reduction of power consumption.

Consider a network which consists of m service providers and n clients, each service provider provides services via a Datacenterbroker B and the clients are required to contact the broker to avail the services. Client forwards his request in the form of cloudlets Vi where 1 <= i <= n and n is the total number of client requests. Each Vi request includes three parameters.

U_i = CPU utilization
M_i = MIPS request
D_i = Deadline

A. Problem of Finding Minimal Cost VM before allocation

Let M_{ik} be the MIPS requested by the k_{th} client for the service. U_{ik} be the CPU utilization expected by the client. D_{ik} be the deadline of cloudlet for both the soft real time and hard real time VM requests. Q_k be the processing element capacity. $Cost_k$ be the cost of provisioning V_i. E_k be the Power consumption of cloudlet. When the cloudlet V_i arrives VM provisioning should be done in such a way that the processing element is not burdened. Therefore Sum of all the $M_{ik} < Q_k$ and $Cost_k$ should be the minimum cost E_k should be the minimum of all the power of V_i

B. *Problem of Reducing VM Sprawl:*

VM provisioning by itself is a manual process where the administrators are required to specify the share or resources to each cloudlet. But nowadays administrators prefer to automate provisioning because manual resource provisioning is often very time-consuming and an error-prone process. In automated process the VMs are created according to a generic template which leads to an uncontrolled proliferation of virtual machines (a phenomenon called VM sprawl).

When a new cloudlet request V_i arrives at the datacenter broker, the datacenter broker can directly allocate it to a VM by creating a specially tailored VM for the cloudlet that just arrived. Therefore if there are N cloudlets request then there needs to be N virtual machines created in hosts. Let V_m.MIPS be the total MIPS allocated to a VM If M_{ik}<V_m.MIPS then a dedicated VM will have only 70% or less resources(MIPS is a special concern here) in use. Therefore the problem here is deciding if creation of new can be avoided and current cloudlet request can be assigned to already running VMs.

C. Problem of Calculating Value of Service

In case of soft real time service model an additional parameter φ is used as a penalty function. Penalty function should decide on the value loss that will be observed if the cloudlet V_i is selected to run on the VM. Penalty function should be used to calculate the profit that will be obtained with and without the cloudlet running in the current VM.

II. LITERATURE SURVEY

Recent trends in cloud computing focus on deploying low cost services to cloud users. JiaRao et.al [1] presented a performance constraint and power-aware allocation for virtual machines using two host selection policies (MAP and MAP-H2L). Their work did not concern reducing the power consumption along with minimum running VMs. Tom et.al [2] described an adaptive provisioning policy for the delivery of cloud resources to cloud theuser applications. Their mechanism makes use of analytical performance or queuing system model. It also intends to improve the queuing model to allow modeling of composite services. Qianet. al[3] proposed a framework for cloud services in a real time environment in which each cloud service request was modeled as Real time Virtual Machine. They have also described about the adaptive and advanced DVFS schemes without their implementations. Rodrigo et.al [4] proposed a mechanism utilizing formulas for calculating energy consumptions in an cloud environment. A power aware scheduling algorithm for applications using DVS scheme has been proposed by the authors of paper [5] for minimizing the energy consumption at the data centers.

III. METHODOLOGY

A. Virtual Machine Real-Time Model

All the requests for the real time VM model can be categorized into two types; Hard real time and Soft real time. In the hard deadline model, a service provider accepts the task only if it can meet the deadline. A service with a soft deadline provides a diminished value or service when the execution time exceeds the deadline which would cost a certain penalty that is calculated here using concave penalty function, convex penalty function and linear decreasing function.

B.Hard Real-Time (HRT) Virtual Machine Provisioning Algorithm (V_i)

The cloudlet HRT-VM,V_i mainly constitutes of three parameters ui, mi, and di where

U_i: the CPU utilization required for the real-time application.

M_i: the number of Millions Instructions Per Second (MIPS) required for the VM.

D_i: the lifetime or deadline.

Therefore a set V which is a collection of all cloudlet requests will comprise of

$V = \{(U_1,M_1,D_1), (U_2,M_2,D_2), (U_3,M_3,M_3),.................(U_k,M_k,D_k)\}$

where 1 <= i <= k and each cloudlet request will be of the form $Vi=(U_i,M_i ,D_i)$.

If a suitable VM was not found for the given request then the execution of that request will be postponed.

3.3 Calculating power consumption & cost of using RAM, Storage, Bandwidth

The dynamic power consumption by a CPU is proportional to V_{dd2} and f, where V_{dd} is the supply voltage and f is the frequency. Since the frequency is usually in proportion to the supply voltage, the dynamic power consumption of a processor is defined by

Dynamic Power Consumption = $k*d_i *S_2$.. (1)

S = mi / Qk... (2)

where**k** is coefficient.

S the relative processor speed for frequency m_i.

Q$_k$the Maximum processor capacity.

Assumptions: k=1

The MIPS of cloudlet is given by MI(Million Instruction)

Price of using I/O bandwidth

Price of Bandwidth = getPricePerSecond() * Total Utilization Of Cpu (ProcessTime())..........(3)

C. Soft Real-Time Virtual Machine Provisioning Algorithm (V_i)

In order to check whether the new cloudlet request V_i can be moved into the VM already created in the host an acceptance test is carried out. The acceptance test calculates the profit for list of all the VMs already running and also the profit for the same list including the new V_i. The difference between the profits is used to decide on the acceptance of the V_i. The penalty function indicates the diminished value of a service by executing a VM that has missed the deadline.

Calculation of delay

delay= finish Time $_m$− dead line $_m$..(4)

Calculation of profit

Profit = Price *Value For Service – Cost : when delay=0..(5)

Profit = Price $_m$(1 – penalty function(delay)) −(finish T_m− t)(Total MIPS/Qk)$_2$

+ Profit(T−{V m},finish Tm); when delay>=0....................................... (6)

Price of using I/O bandwidth

Price of Bandwidth = get Price Per Second() * TotalUtilizationOfCpu(ProcessTime()).......... (7)

Calculation of value for service

Value For Service = { 1 if delay = 0; 1-penaltyfunction(delay); if delay>0
0 if delay >= deadline.
.... (8)

Penalty functions implemented:

Linear Penalty function = (1/deadline)*delay... (9)

Convex Penalty function = (delay / deadline2) * delay.. (10)

Cost calculation using DVFS equation:

Cost of Power = k * (finish Tm −t)(Total MIPS/Q_k)^2...(11)

Acceptance Test:

Profit current = Profit(T, current Time)...(12)

Profit new = Profit (T, current Time)... (13)

Profit k = Profit new – Profit cur.. (14)

MIPS Calculation for the acceptance test:

MIPSj = wj/(dj − t)..(15)

TotalMIPS = TotalMIPS + MIPSj.. (16)

finishTimej = wj / MIPSj..(17)

D. Basic Idea of the Algorithm

Step 1 : Designate a server as a broker within a datacenter.
Step 1 : Check for necessary parameters in Current cloudlet request.
Step 2 : Get MIPS, deadline and CPU utilization of cloudlet expected.
Step 3 : Get the list of already running VM's.
Step 4 : Calculate already allocated MIPS in VM.
Step 5 : Get the Total MIPS of a virtual machine in a host.
Step 6 : Check if the available MIPS is equal to requested MIPS
Step 7 : Calculate power of current cloudlet in VM.
Step 8 : Calculate cost of current cloudlet in VM.
Step 9 : .Select a minimally priced and power VM.
Step 10 : In case of a tie, VM which consumes minimum energy is selected.
Step 11 : In case of delay, calculate profit to check for acceptance of cloudlet.
Step 12 : Return the VM
Step 13 : Allocate cloudlet to VM
Step 14 : If VM not available, create a new VM for current cloudlet request.
Step 15 : Return new VM
Step 16 :

IV. EXPERIMENTATION AND EVALUATION

The experimentation is carried out through a simulation toolkit called "Cloudsim", because evaluation of the implementation in real time environment is highly expensive. Considering the same reason a datacenter environment is simulated using Cloudsim toolkit. The algorithm is experimented using Planetlab workload.

Status	Cloudlet ID	Datacenter ID	VM ID	Time (sec)	Start Time (sec)	Finish Time (sec)
SUCCESS	105	3	102	1.2	200.1	201.3
SUCCESS	102	3	104	1.2	200.1	201.3
SUCCESS	107	3	101	1.6	200.1	201.7
SUCCESS	101	3	100	2	200.1	202.1
SUCCESS	108	3	101	2.2	200.1	202.3
SUCCESS	103	3	103	2.5	200.1	202.6
SUCCESS	106	3	103	2.65	200.1	202.75
SUCCESS	100	3	100	2.76	200.1	202.86

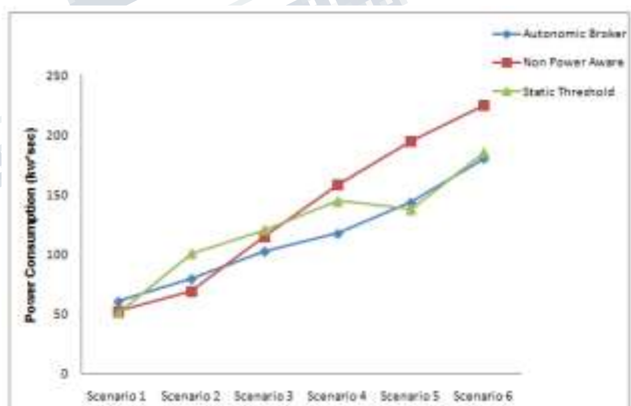
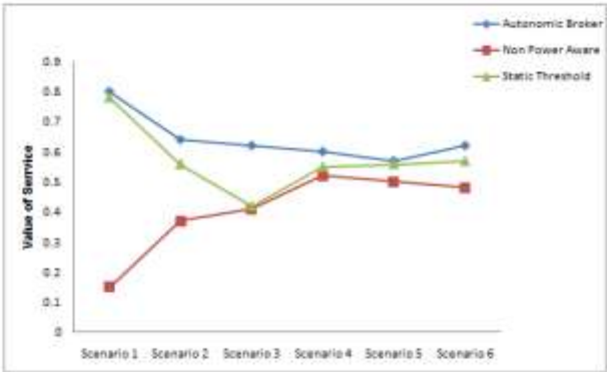


Table 1: Status of VM creation Figure 1: Power consumption of Hard Real time Cloudlet.



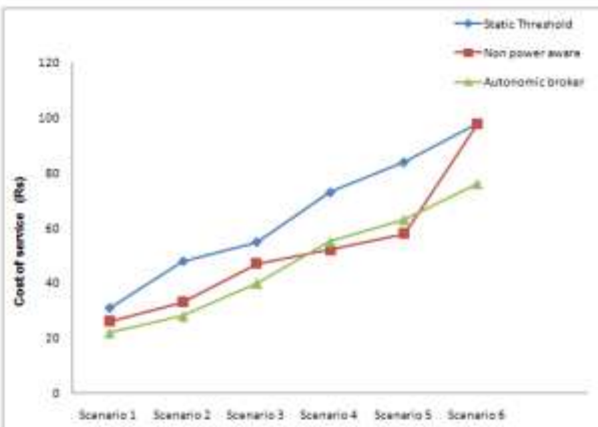


Figure 2: Value of service graph for Soft Real Time Cloudlet Figure 3 : Cost graph for Soft Real Time Cloudlet Request

In table 1 the status field denotes the status of the allocation of cloudlet that arrives. If suitable virtual machine running exists which satisfies both the user specified requirements as well as the requirements of power and cost of the algorithm then the VM is selected and cloudlet is assigned to same VM otherwise new VM created. If creation of VM is not possible then the allocation of the cloudlet is postponed and the status of allocation of cloudlet is failure. Second main intention of the proposed algorithm is to reduce VM sprawl, as seen in the table 1 cloudlet 100 and cloudlet 101 are assigned to same Virtual machine with ID 100. Similarly cloudlet 107 and 108 are held in VM 101. Fig. 1 shows that theautonomic broker based mechanism consumes less power compared to other algorithms. Fig. 2depicts clearly that the value of service is higher than other algorithms. Fig. 3 shows the cost of processing for each algorithm considered and clearly describes the considerable reduction of cost for the proposed autonomic broker based mechanism.

V. CONCLUSION

Cloud computing is a model which allows applications or programs which are hosted in remote servers to be run and shared by different user clients at the same time using virtualization. The proposed work focuses on improving resources consolidation by considering themetrics like power, cost and the QoS.QoS is ensured by checking the percentage of SLA violation. A power-aware provisioning of VMs for soft and hard real-time Cloud services is run using DVFS schemes resulting in considerable reduction in power consumption and VM sprawling. For hard real-time requests a minimally priced virtual machine is returned. This Virtual Machine returned takes into consideration both the price of processing and the power consumption. Thus it makes sure that a new VM is created only if the existing VMs that are running cannot accommodate the new cloudlet. This way VM creation is kept in control and resource utilization is minimized.For soft real-time requests algorithm chooses the VM which would have the maximum profit. The simulation results have shown that datacenters can reduce power consumption and increase their profit using DVFS schemes. The penalty calculations for soft real time using linear decreasing function and concave penalty function have shown improved QoS without affecting the overall power consumption and the cost.

REFERENCES

[1] JiaRao, Jiayu Gong and Cheng-ZhongXu, “QoS Guarantees and Service Differentiation for Dynamic Cloud Applications”, IEEE Transactions on Network And Service Management, Vol. 10, No. 1, March 2013.

[2] Tom Guerout, Thierry Monteil, Georges Da Costa, Rodrigo NevesCalheiros, RajkumarBuyya, MihaiAlexandru, “Energy-aware simulation with DVFS”, 2013 Elsevier B.V, 30 April 2013.

[3] Qian Zhu and GaganAgrawal, “Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments”, IEEE Transactions On Services Computing, Vol. 5, No. 4, October-December 2012.

[4] Rodrigo N. Calheiros, Rajiv Ranjanyand RajkumarBuyya, “Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments”, IEEE computer society,2011.

[5] A. Beloglazov, J. Abawajy and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing”, Future Generation Computer Systems (FGCS), vol. 28, no. 5, pp. 755–768, 2011.

[6] 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..

[7] Salvatore Distefano, Politecnico di MilanoAntonioPuliafito, "A new Cloud@Home paradigm aims at merging the benefits of cloud computing".