ICICES2014 - S.A.Engineering College, Chennai, Tamil Nadu, India

Profit and Energy Aware Scheduling in Cloud Computing using Task Consolidation

Ms.A.Bharathi
PG Scholar
Department of CSE
Kongu Engineering college,
Erode-638052, India
Email:bharathirari@ymail.com,
Mobile No:9262403684

Mrs.R.S.Mohana
Assistant Professor
Department of CSE
Kongu Engineering College
Erode-638052, India
Email:mohanapragash@kongu.ac.in
Mobile No:8012187005

Ms.A.Ushapriya
Assistant Professor
Department of CSE
Vivekananda Institute of Engg & tech for women, Tiruchenkode-637205, India
Email:usha_ppm2000@yahoo.com
Mobile No:7845174942

***Abstract*—Cloud computing systems rent resources on demand, pay-as-you-go basis, and multiplex many users on the same physical infrastructure. However the revenue of cloud computing is get affected by various factors such as QoS constraints, Energy consumption etc., Energy Aware Task Consolidation technique is used to allocate the tasks dynamically on virtual clusters which aims to minimize energy consumption. This is achieved by consolidating tasks on virtual clusters by keeping the CPU utilization below a peak threshold value of 70%. The task consolidation is done by using bestFit strategy. The revenue of cloud provider can be improved by increasing the profit yielded by the incoming task. The profit can be increased by allocating the task to the appropriate VM which executes the task with minimum cost and without violating the QOS constraints. In this work, Profit and Energy aware Task Consolidation method is proposed to allocate the task to the appropriate VM that yields more profit and less energy consumption to the data center.**

***Index Terms*—Cloud computing, Energy consumption, Task consolidation, Profit model, Scheduling.**

I. INTRODUCTION

Cloud computing has become popular due to the maturity of related technologies such as network devices, software applications and hardware capacities. Resources in these systems can be widely distributed and the scale of resources involved can range from several servers to an entire data center. To integrate and make good use of resources at various scales, cloud computing needs efficient methods to manage them. Consequently, the focus of much research in recent years has been on how to utilize resources and how to reduce power consumption.

Resource allocation is one of the most important and difficult tasks in Cloud systems. It is the task of spreading a finite group of resources across a user population and it forms the basis of modern economics.

The problem of resource management is considered for a large scale Cloud Environment which includes the physical infrastructure and associated control functionality that enables the provisioning and management of cloud services.

Until recently, high performance has been the sole concern in data center usage, and this demand has been fulfilled without giving much attention to energy consumption.

The larger deployment of resources in cloud environment leads to a greater energy consumption. An average data center consumes as much energy as 25,000 households. As energy costs are increasing while availability shrinks, there is a need to shift the focus from optimizing data center resource management for pure performance to optimizing them for energy conservation, while maintaining high service level performance.

One of the key technologies in cloud computing is virtualization. The ability to create virtual machines (VMs) dynamically on demand is a popular solution for managing resources on physical machines. Therefore, many methods have been developed that enhance resource utilization such as memory compression, request discrimination, defining threshold for resource usage and task allocation among VMs. Improvements in power consumption, and the relationship between resource usage and energy consumption has also been widely studied . Some research aims to improve resource utilization while others aim to reduce energy consumption. The goals of both are to reduce costs for data centers. Due to the large size of many data centers, the financial savings are substantial.Energy consumption varies according to CPU utilization. Higher CPU utilization usually implies greater energy consumption. However, higher CPU utilization does not equate to energy efficiency. This phenomenon motivates the idea of not exhausting CPUs with high levels of utilization (for example, 80–100%) in order to save energy.

This work contributes towards engineering a middleware layer that performs resource allocation in a cloud environment, ith the goal of increasing the revenue obtained by the data centers without affecting performance of cloud environment.

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

The cloud can become more popular by increasing profit and reducing energy consumption without any degradation of service level requirements mentioned by the users.

II. RELATED WORK

When cloud environment is concentrating on increasing profit, Florentina et al (2005) proposed an algorithm composed of many different scheduling policies which are mainly based on increasing of profit without any degradation of quality of service. Using economic based approach many scheduling policies are developed which will systematically allocate the jobs to the resource with concern of profit. In this work FirstOpportunity and FirstOpportunityRate scheduling model provide a better profit gain compared to the FirstProfit scheduling algorithm.

Young et al (2012) proposed two algorithms called ECTC and MaxUtil which aims to maximize resource utilization and explicitly take into account both active and idle energy consumption. It will assign each task to the resource on which the energy consumption for executing that task is reduced both implicitly and explicitly without degradation of performance of that task.

Profit is more considerable factor in cloud computing system after development of cloud environment. Young et al (2010) proposed two profit based algorithms called MaxProfit and MaxUtil algorithm where MaxUtil algorithm selects an instance with low utilization and MaxProfit algorithm selects a task with earliest start from the queue which is maintained by MaxUtil technique.

Zheng et al (2011) proposes a novel technique for load balancing from the power perspective and targets at operating servers in a reasonable range of utilization levels, for example between 30% and 70%. From the service provider's point of view, the goal of load balancing algorithm was to reduce power consumption and to earn more profit. To achieve this, two pricing algorithms for load distribution was designed. Both algorithms considered utilization of computers besides factors, such as pricing and power cost. In the first algorithm, pricing functions with respect to the computer utilization based on the resource usage was designed. In the second algorithm, profit that a service provider earned after deducting power cost from its revenue was focused.

Cloud computing require that both customers and providers should be confident that signed SLA are supporting their respective business activities to the best extent. The confidence is not provided by currently used SLAs, especially when providers outsource resources to other providers. These resource providers support very simple metrics like availability, or metrics that prevent an efficient exploitation of their resources. Goiri et al (2011) used a resource-level metric for specifying fine-grain guarantees on CPU performance. This metric allowed resource providers to allocate dynamically their resources among running services based on their demand. This is completed by incorporating the customer's CPU usage in the metric definition, and avoiding fake SLA violations when the customer's task does not use all its allocated resources. Resource level metric violates fewer SLA than other CPU related metrics and QoS is maintained.

Lien et al (2006) proposed a method for measuring the power consumption of a streaming media server without any additional hardware meter which is highly depends on the characteristics of its load. The power consumption is highly related with the CPU utilization and it varies according to the load level of CPU utilization. The software method proposed in this work measures power consumption based on the accessing of the real time CPU utilization.

III. ENERGY MODEL

Energy consumption and CPU utilization will not increase linearly. Energy consumption of a virtual machine is defined in many levels of CPU utilization. Energy consumption varies in accordance to the CPU utilization. In idle stage of CPU utilization, it consumes some α w/s amount of energy. And it increases to some β w/s amount of energy while the CPU utilization increases. Thus the energy consumption of a VM V_i is defined as follows:

$$E(V_i) = \begin{cases} \alpha \text{ w/s}, & \text{if idle} \\ \beta + \alpha \text{ w/s}, & \text{if } 0\% < \text{CPU utilization} \le 20\% \\ 3\beta + \alpha \text{ w/s}, & \text{if } 20\% < \text{CPU utilization} \le 50\% \\ 5\beta + \alpha \text{ w/s}, & \text{if } 50\% < \text{CPU utilization} \le 70\% \\ 8\beta + \alpha \text{ w/s}, & \text{if } 70\% < \text{CPU utilization} \le 80\% \\ 11\beta + \alpha \text{ w/s}, & \text{if } 80\% < \text{CPU utilization} \le 90\% \\ 12\beta + \alpha \text{ w/s}, & \text{if } 90\% < \text{CPU utilization} \le 100\% \end{cases}$$

IV. TASK CONSOLIDATION

The task's that are arrived to the data center will be accepted from the user with the information which is used to calculate the profit of the tasks.

The information that is required to process the task is shown in Table 1.

TABLE 1. List of Tasks

Task t_i	Arrival time($t_{a,i}$)	Processing time($t_{p,i}$)	Cpu utilization	Data size
t_0	0s	50s	30%	150 Mb
t_1	10s	20s	30%	75 Mb
t_2	12s	35s	40%	20 Mb
t_3	15s	15s	30%	150 Mb
t_4	20s	30s	60%	250 Mb
t_5	30s	25s	30%	110 Mb
t_6	35s	10s	50%	210 Mb

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

When the task is entered into the data center, it will be consolidated to the VM's in the current VC. The tasks are scheduled only up to the 70% CPU utilization. If it bypasses the threshold value, then the nearer VM which is not exceeding 70% CPU utilization will be selected for executing task. If none of the VM in the current VC is capable of executing incoming task then the Neighbor VC with low cost of transmitting the task will be selected.

A. Transfer Cost Calculation

The cost for transferring data from one VC to another VC is calculated as in Equation (1):

$$Cost_{i,j} = \sum_{t=T_{a,j}}^{T_{a,j}+T_{p,j}} E_t(V_i) + DS_j / BW_{PQ} \times 2\beta \text{ W/s} \quad (1)$$

where,

DS_j = Data size of current task,

BW_{PQ} = Bandwidth from current VC to neighbor VC

β = Energy consumed at particular CPU utilization

V. PROFIT MODEL

The goal of profit model is to maximize the profit earned by the data centers in accordance to the consideration of quality of service parameters of the tasks mentioned by the users. Before allocating tasks to the VM's the cost of executing that task will be calculated by using profit model.

A. Profit Calculation

The profit model is used to calculate the cost incurred for executing each and every task's in the VM. Let $Cost_{ijl}^{new}$ be the total cost incurred to the data center for processing the user request on VM i type l and virtual cluster j. Then, the profit ($Prof_{ijl}^{new}$) gained by the data center is defined in Equation (2):

$$Prof_{ijl}^{new} = B^{new} - Cost_{ijl}^{new}; \forall i \in I, j \in J, l \in N_j \quad (2)$$

where,

B^{new} = User budget,

$Cost_{ijl}^{new}$ = Total cost for processing task

The total cost incurred to data center for accepting the new request on VM i of type l is calculated with the variance of processing cost, data transfer cost, initiation cost and penalty delay cost. Thus the total cost for processing incoming task on type l of VC is given in Equation (3):

$$Cost_{ijl}^{new} = PC_{ijl}^{new} + DTC_{jl}^{new} + IC_{ijl}^{new} + PDC_{ij}^{new}; \forall i \in I, j \in J, l \in N_j \quad (3)$$

where,

PC_{ijl}^{new} = Request's processing cost,

DTC_{jl}^{new} = Data Transfer Cost,

IC_{ijl}^{new} = VM initiation Cost,

PDC_{ij}^{new} = Penalty delay cost.

The processing cost for serving the request is dependent on the new request processing time and hourly price of VM_{il}. Thus PC_{ij}^{new} is given in Equation (4):

$$PC_{ijl}^{new} = procT_{ijl}^{new} \times P_{jl}; \forall i \in I, j \in J, l \in N_j \quad (4)$$

where,

$procT_{ijl}^{new}$ = New request processing time,

P_{jl} = Hourly price of VM for execution.

The Data transfer cost includes cost for both data-in and data-out. Thus the DTC is given in Equation (5):

$$DTC_{jl}^{new} = inDS^{new} \times inPri_{jl} + outDS^{new} \times outPri_{jl}; \forall j \in J, l \in N_j \quad (5)$$

where,

$inDS^{new}$ = Data-in required to processing the user request,

$inPri_{jl}$ = Price charged for data transfer-in,

$outDS^{new}$ = Data-out required to processing the user request,

$outPri_{jl}$ = Price charged for data transfer-out.

The initiation cost of VM i is dependent on the type of VM initiated in the data center. Thus the IC can be calculated as given in Equation (6):

$$IC_{jl}^{new} = iniT_{ij} \times P_{jl}; \forall i \in I, j \in J, l \in N_j \quad (6)$$

where,

$iniT_{jl}$ = Time taken for initiating VM I of type I,

P_{jl} = Hourly price of VM to process task.

Penalty delay cost id how much the service provider has to give discount to users for QOS violation. It is dependent on penalty rate and penalty delay time period. The PDC can be calculated by using Equation (7):

$$PDC_{ijl}^{new} = \beta^{new} \times PDT_{ijl}^{new}; \forall i \in I, j \in J, l \in N_j \quad (7)$$

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

where,

β^{new} = Penalty rate,

PDT_{ijl}^{new} = Penalty delay time.

To process any new request, data center either can allocate a new VM or schedule the request on an already initiated VM. If it schedules the new request on an already initiated VM_i, then the new request has to wait until VM I becomes available. The time taken by new request to wait until it starts processing on VM I is $\sum_{k=1}^{K} procT_{ijl}^{k}$, where K is the number of request yet to be processed before the new request. Thus PDT_{ijl}^{new} is given by Equation (8):

$$PDT_{ijl}^{new} = \begin{cases} t+\sum_{k=1}^{K} procT_{ijl}^{k}+procT_{ijl}^{new}-DL^{new}, \text{if new VM is not initiated} \\ procT_{ijl}^{new}+iniT_{ijl}+DTT_{ijl}^{new}-DL^{new}, \text{if new VM is initiated} \end{cases} \quad (8)$$

where,

t = Time instant,

$\sum_{k=1}^{K} procT_{ijl}^{k}$ = Waiting time of new request to be processed,

$procT_{ijl}^{new}$ = Processing time,

DL^{new} = Task deadline,

$iniT_{ijl}$ = Time for initiating VM,

DTT_{ijl}^{new} = Data transfer time

DTT_{ijl}^{new} Is the data transfer time which is the summation of time taken to upload the input and download the output data from the $VM_{il.}$ The DTT is defined in Equation (9):

$$DTT_{ijl}^{new} = inDT_{ijl}^{new} + outDT_{ijl}^{new}; \forall i \in I, j \in J, l \in N_j \quad (9)$$

where,

$inDT_{ijl}^{new}$ =Time taken to upload input,

$outDT_{ijl}^{new}$ =Time taken to download input.

Thus, the response time for the new request to be processed on VM is given as in the Equation (10):

$$T_{ijl}^{new} = \begin{cases} \sum_{k=1}^{K} procT_{ijl}^{k}+procT_{ijl}^{new}, \text{if new VM is not initiated} \\ procT_{ijl}^{new}+iniT_{ijl}+DTT_{ijl}^{new}, \text{if new VM is initiated} \end{cases} \quad (10)$$

The investment return to accept new user request per hour on a particular VM is calculated as in the Equation (11):

$$ret_{ijl}^{new} = \frac{prof_{ijl}^{new}}{T_{ijl}^{new}}; \forall i \in I, j \in J, l \in N_j \quad (11)$$

B. Algorithm

The Profit and Energy Aware Task Consolidation (PETC) algorithm is works as follows:

Algorithm: PETC

1. Check_Threshold ($VC_{current}$, t_j)
2. **NoVC:**
3. **If** VM's in VC can execute task without surpassing the CPU utilization threshold
4. {
5. Calculate_Cost (t_j)
6. Consolidate_Task (VM , t_j)
7. **If** no VM is appropriate to execute that task then
8. **Goto** Next
9. **Else**
10. Calculate energy consumed
11. }
12. **Next:**
13. **Else**
14. {
15. Check_Threshold ($VC_{neighbor}$, t_j)
16. **If** more than one VC available with required space, then
17. Calculate_costTransmission (BW , DS)
18. **Else**
19. **Goto** NoVC
20. }

The flow of process while trying to allocate tasks with concern of reducing energy consumption is shown in Figure 1.

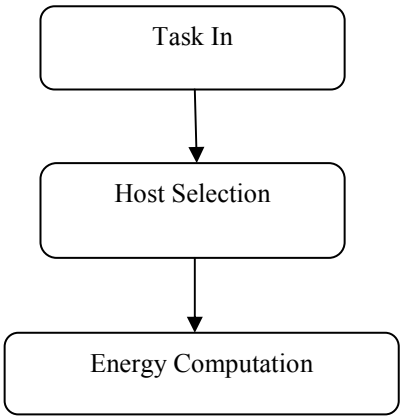

Fig 1. Work Flow Diagram

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

The host selection process with the focus of increasing profit is shown in Figure 2

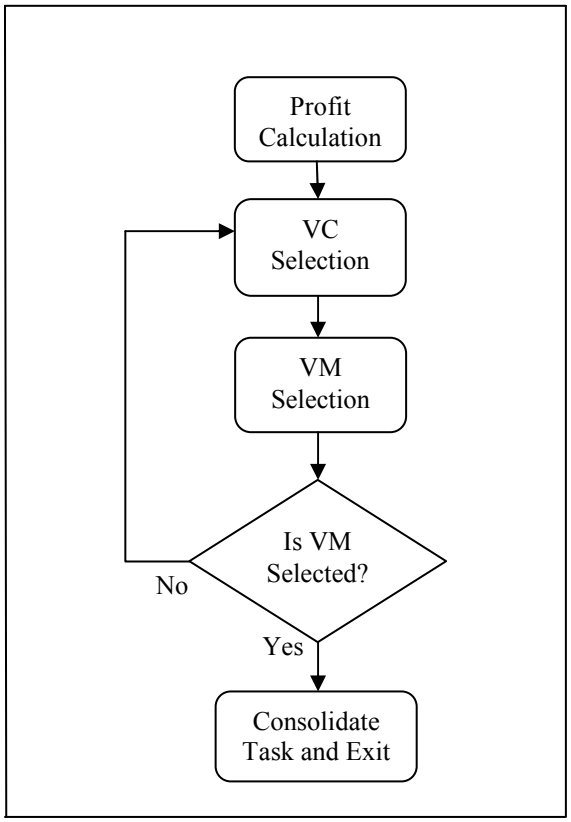

Fig 2. Host Selection

VI. EVALUATION OF PERFORMANCE

To evaluate the performance of the proposed technique, ETC method and PETC method is proposed. ETC method consolidates tasks into the VM only up to the 70% CPU utilization. PETC method consolidates the tasks based profit of executing the tasks as well as minimum energy consumption. Figure 3 shows the result obtained with the comparison of ETC method and PETC method. The results are compared based on different number tasks with profit achieved by executing those tasks.

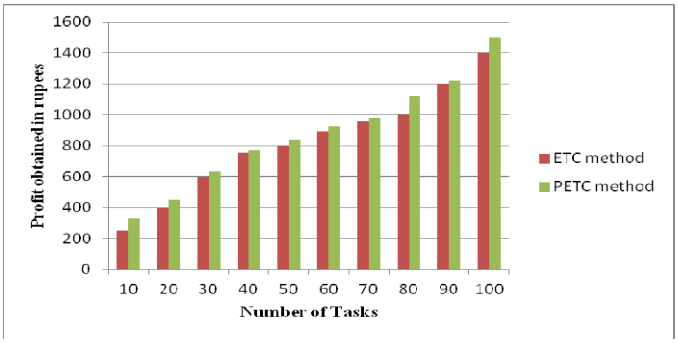

Fig 3. The results with different number of tasks

VII. CONCLUSION

In the cloud environment, the revenue of cloud data centers is affected by various parameters. The main factor that degrades the revenue of data center is energy consumed for executing each and every task's. The ETC algorithm is proposed to schedule the tasks to the appropriate resources in which the energy consumption of executing that task is less using task consolidation technique.

The system is designed to consolidate the tasks that are entering into the data centers. The tasks are consolidated up to the threshold value 70% CPU utilization in order to reduce energy consumption. Though the requested virtual cluster is not enough to execute the task, the system will allocate the task to another virtual cluster by considering energy consumption.

In order to increase profit, the tasks are allocated to the data centers based on the profit model. The goal of profit model is to maximize the profit earned by the data centers in accordance to the consideration of quality of service parameters of the tasks mentioned by the users. Before allocating tasks to the VM's the cost of executing that task will be calculated by using profit model.

VIII. REFERENCES

[1] Goiri I., Julia F., Oriol Fito J., Macias M. and Guitart J. "Supporting CPU based guarantees in cloud SLAs via resource-level QoS metrics", Future Generation Computer Systems 28, 2011, pp.1295–1302

[2] Lien C.H., Liu M.F., Bai Y.W., Lin C.H. And Lin M.B. "Measurement by the software design for the power consumption of streaming media servers", Instrumentation and measurement technology conference,2006, pp.24-27

[3] Lee Y.C. Wang C., Zomaya Y. and Zhou B.B., "Profit-driven service request scheduling in clouds", Proc.IEEE/ACM conf. cluster, cluster and grid computing, 2010, pp.15–24

[4] Lee Y.C. Wang C., Zomaya Y. and Zhou B.B., "Profit-driven scheduling for cloud services with data access awareness", J.Parallel Distrib. Comput. 72, 2012, pp. 591–602

[5] Lee Y.C. and Zomava Y., "Energy efficient utilization of resources in cloud computing systems", The Journal of super computing 60, 2010, pp.268–280

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE

[6] Popvici F.I. And Wilks J., "Profitable services in an uncertain world", the ACM/IEEE conf. Hign performance Networking and computing, 2005.

[7] Wu L., Garg S.K. and Buyya R., "SLA-based admission control for a Software-as-a-Service provider in cloud computing environments", Journal of Computer and System Sciences 78, 2012, pp.1280–1299

[8] Zheng Q. and Veeravalli B., "Utilization-based pricing for power management and profit optimization in data centers", J. Parallel Distrib. Comput.72, 2011, pp. 27–34

ISBN No.978-1-4799-3834-6/14/$31.00©2014 IEEE