Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 1 (2017) pp. 1-9
© Research India Publications
http://www.ripublication.com

Energy Aware Resource Allocation for Data Center

Nikita Patel
PG student[1]

Prof. Nimisha Patel
Associate Professor[1] *Research Scholar*[2]

Dr. Hiren Patel
Professor[3]

[1] *Sankalchand Patel College of Engineering Visnagar-Gujarat, India*
[2] *Rai University, Ahmedabad-Gujarat, India*
[3] *LDRP Institute of Technology & Research, Gandhinagar-Gujarat, India*

Abstract

Cloud computing provides on-demand computing resources based on pay-as-you-use model on rental basis. In last few years, Cloud computing has come up with wide adaptation and ease of use, due to which gigantic number of datacenters across globe has created an issue of energy consumption and carbon emission. Distribution of workload among available virtual resources of datacenter is one of the major concerns in addressing the issue of energy consumption which can be handled with proper resource allocation. Effective resource utilization helps maintaining load balancing and improves system performance. In this paper, our work is mainly targeted on dynamic allocation of virtual machines (VMs) to the user tasks, based on analyzing characteristics of task (such as task's deadline and execution time) for maximum utilization. Alternatively, low priority task should not delay the execution of high priority and accordingly, allocate the resources dynamically for task within deadline. We also target to contribute in priority-based mapping VM on hosts.

Keywords: Cloud computing, Resource Allocation, Task Scheduling

1. INTRODUCTION

The Cloud is a network of servers which are used for many purposes including sharing computing resources, running applications and data storage. It offers services in various flavor viz. Infrastructure as a Service, Platform as a Service, and Software as a Service. Cloud computing offers pay-as-you-use model where computing resources (such as memory, processing elements, bandwidth, storage etc.) are provisioned over the Internet based on the user demand with elasticity and flexibility.[1] Virtualization is key concept for Cloud to allow scale up and down their resources based on needs. Virtual resources are allocated to the user based on several

factor of tasks to improve the utilization of resources and to reduce energy consumption.[2] Increase usage of Cloud leads to increase in number of data centers which in turn results into raise in energy consumption. Efficient resource allocation has been identified as one of the significant domains to address the issue of energy consumption.

Resource allocation is a process of providing required resources for said duration during requested time by the user for a given task. All submitted tasks are stored in a queue. Scheduler running on dedicated system manages all tasks and pool of resources, and decides whether to provision new VM from Cloud and/or to allocate task to VM.[3] In this work, we aim to address dynamic allocation of virtual machines (VMs) to the user tasks, based on analyzing characteristics of task (such as task's deadline and execution time) for maximum utilization. Alternatively, we do not want the low priority task to delay the execution of high priority and consequently, allocate the resources dynamically for task within deadline. We also target to contribute in priority-based mapping VM on hosts.

The reminder of this paper is structured as follows: In section 2, we describe related work about the resource allocation. Section 3 describes our proposed algorithm. Our conclusion and references are described in Section 4 and 5, respectively.

2. RELATED WORK

In Cloud, virtual resources are allocated for effectively handling workload fluctuations, while providing Quality of Service (QoS) to the end users. The computing and network resources are limited and have to be efficiently shared among the users in virtualized way. In order to perform effective resource management, we need to consider the issues such as resource mapping, resource provisioning, resource allocation, resource adaption and resource scheduling. Resource adaptation is the capacity to adjust the resources dynamically to fulfill the requirements of the user. Resource provisioning is the allocation of a service provider's resources to a user. Resource scheduling is a timetable of events and resources. It determines when an activity should start or end, depending on duration, predecessor activities, predecessor relationships, and resources allocated. Resource mapping is a correspondence between resources required by the users and resources available with the provider.[4]

Proper utilization of available resources is a challenge for energy aware resource allocation in Cloud datacenter. While allocating a task to a resource, few questions need to be answered: How to allocate resources to tasks? What should be the order of execution for given tasks in Cloud? How to overcome overheads when VMs create, terminate or switch tasks? [2] We aim to address the above mentioned issues by proposing an efficient resource allocation and task scheduling in Cloud.

Bagheri at el [2] present technique for energy-aware resource provisioning in real time

tasks in Cloud. Since, the primary requirement of such services are based on deadline constraints, several policies for provisioning of VMs and hosts are allocated in order to decrease energy consumption and miss rate of deadline by an increase in VMs density on hosts and switching the idle ones off. Here, VM are distributed among the hosts under the energy–aware resource allocation policy which try to reduce minimum number of active hosts. Here, resource allocation is claimed to be optimized in order to increase the acceptance rate of real time tasks through VM scaling and migration.

Saraswathi at el [3] propose a method for execution of high priority tasks. Here, creation of new VM for newly arrived task is avoided. It leads to resource contention between low and high priority tasks to access resources. The main contribution of its work is priority-based preemption policy that improves resource utilization in virtual environment.

Pawar at el [5] focus on dynamic resource allocation mechanism for preemption jobs in Cloud. Proposed algorithm dynamically responds to fluctuating workload through preempting the current running task having low priority with high priority task and if preemption is not possible because of same priority then by creating new VM from globally available resources. If global resources are not available, task will be placed in waiting queue. When appropriate VM becomes free that advanced reservation task will be selected from waiting queue and allocated for execution to that VM. They propose priority based algorithm, which considers multiple SLA parameter and resource allocation by preemption mechanism for high priority task execution by best-effort job it will improve utilization in Cloud.

Goutam at el [6] work on dynamic resource provisioning. They present a scheduling heuristic considering multiple SLA objectives, such as amount of required CPU, network, bandwidth, and cost for deploying applications in Clouds. They present a local and global scheduling based on user's service request and also present novel method for high priority task. It is also beneficial for fault tolerance procedure in resource management. If a resource is going to get failed then it is immediately allocated with new resource for task. In this algorithm the priority of task is considered over cost and deadlines.

Çalar at el [7] propose an adaptive decision making approach in order to achieve energy-efficient allocation without VM migration. Authors contribute on VM migration while considering remaining time of running workloads. Optimum utilization is a main factor in order to provide energy-efficiency. Allocation of workload to an already active host will increase the optimum utilization rate instead of allocation the workload to a new servers would be preferred. They considered not only history but also future demands and remaining time of running workloads.

3. PROPOSED WORK

Our Propose work has been mainly categorized among two phases:

- A: Mapping of resources (VMs) on hosts.
- B: Mapping of tasks (Cloudlet) on resource (VM).
  - B1.1: Priority based mapping.

For part A, we propose a technique to allocate VMs on hosts based on capacity of host, as mentioned in Algorithm 1.

Algorithm 1: Allocation of VM on host based on capacity of host

```
//Allocation of VM on HOST
Input: vmList, hostList
Output: resourceAllocationMap
 //bifurcate host based on processing capacity
1.  Capacity ←High
2.  for each host in hostList do
3.      if host.getCapacity() > Capacity then
4.          hostListHigh.add(host)
5.      else
6.          hostListModerate.add(host)
7.     end if
8.  end for
9.  for each vm in vmList do
10.     if vm.getPriority()=HIGH then
11.          remainingMIPS←MAX
12.         for each host in hostListHigh do
                //check active/inactive host
13.               if host.getStatus()=ACTIVE then
14.                if host has enough resources for VM then
15.                 if host. getUtilization () + vm.getUtilization ()< UpperThreshold then
16.                   if UpperThreshold-(host.getUtilization()+vm.getUtilization()) <
                          remainingMIPS  then
17.                        remainingMIPS← UpperThreshold-(host.getUtilization()
                           +vm.getUtilization())
```

```
18.                         allocatedhost←host
19.                     end if
20.                   end if
21.                end if
22.             end if
23.          end for
24.          If allocatedhost is null then
25.               activate a new suitable host from hostlistHigh
26.               allocatedHost←host
27.          end if
28.       else
29.          find suitable host for placing vm
30.          allocatedHost←host
31.       end if
32.       If allocatedHost is not NULL then
33.            resourceAllocationMap.add (vm,allocatedHost)
34.       end if
35. end for
```

Initially, based on processing capacity, the available hosts are classified among two groups viz. (i) hosts with high processing capacity and (ii) hosts with moderate processing capacity. For every VM with high priority, list of hosts with high processing capacity are checked for following conditions: (a) The host should be active i.e. already in use. (b) The host should have enough resources to serve the VM (c) After placing the VM on the host, the host should not get overloaded (d) The VM should be placed on a host in such a way that the resultant utilization reaches nearer to upper threshold. For all these conditions, if we find a host, then we place the VM on it. Otherwise, we activate a sleeping host with high processing capacity and place the VM on this newly activated host. Following figure 1 depicts the diagrammatic representation in form of flowchart of the algorithm.

Flowchart

Figure 1 it's shows the flowchart of Allocation of VM on host based on capacity of host.

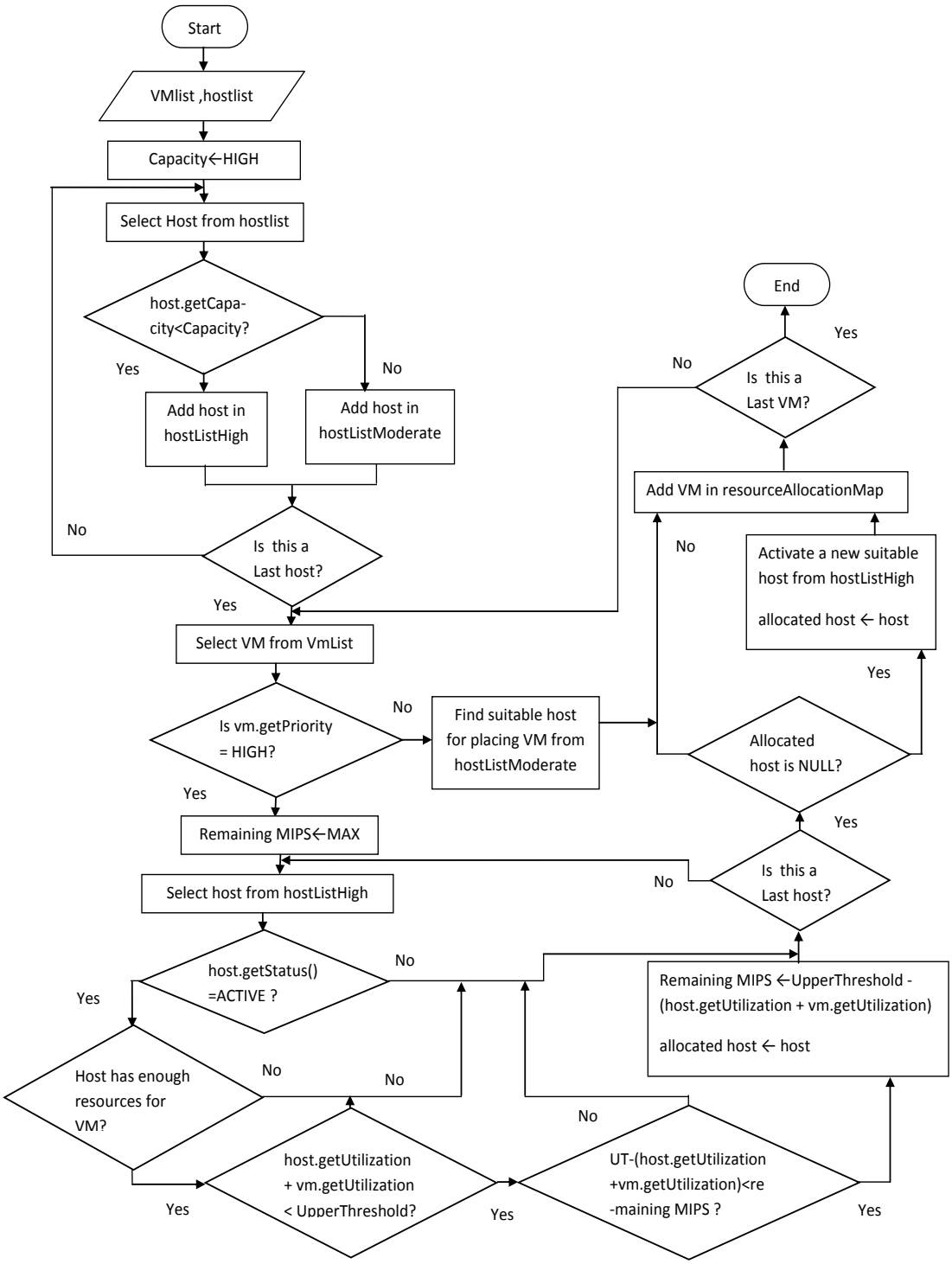

Figure 1. Flowchart of proposed algorithm 1.

For Part B, we identified gap in Existing research [3], it compares task time (i.e. overall deadline) of conflicting tasks and allows the task with lower size. But, we recommend comparing the size of newly arrived task with the remaining time of already executing task.

Further, we recommend that, if size of both the tasks happens to be same, already executing task should be allowed to continue. We propose a dynamic VM allocation model based on the characteristics of the task such as deadline and execution time, which can dynamically reconfigure virtual resources and thereby increasing the resource utilization. Following algorithm illustrate the steps for the same.

Algorithm 2: Execution of High priority Task when all existing resources are allocated

Input: New task, all tasks running in host
Output: Execution of all tasks submitted to the host

1. Begin
2. Arrival of New task
3. If (New task. deadline< all tasks running in host)
4. High priority task = New task
5. If (VM is available)
6. Allocate High priority task to that VM
7. else if (Running task. Remaining execution time > New task. execution time)
8. Suspend task ←Selection of task for execution of high priority task();
9. Suspend (Suspend task)
10. Allocate High priority task to VM from which a task was suspended
11. else
12. Execute running tasks in VM
13. End if
14. Execution of all tasks in VM
15. If (completion of a task which is running in VM)
16. Resume (Suspend task)
17. Allocate the resumed task to that VM
18. End if
19. Execution of resumed task
20. End

For a newly arrived task, first we check for priority. If the task is of high priority, then we seek for available VM for mapping the task on it. If VM is not available then we need to suspend an already running task. Selection of the task to be suspended depends on the remaining execution time of already running task. If it is more than the required execution time of new task, then we suspend the running task and schedule the new task on that VM. If it is not the case, we let the task continue and wait till its completion. After completion of new task's execution, we resume the suspended task.

Flowchart

Figure 2 it's shows the flowchart of execution of high priority task when all existing resources are allocated.

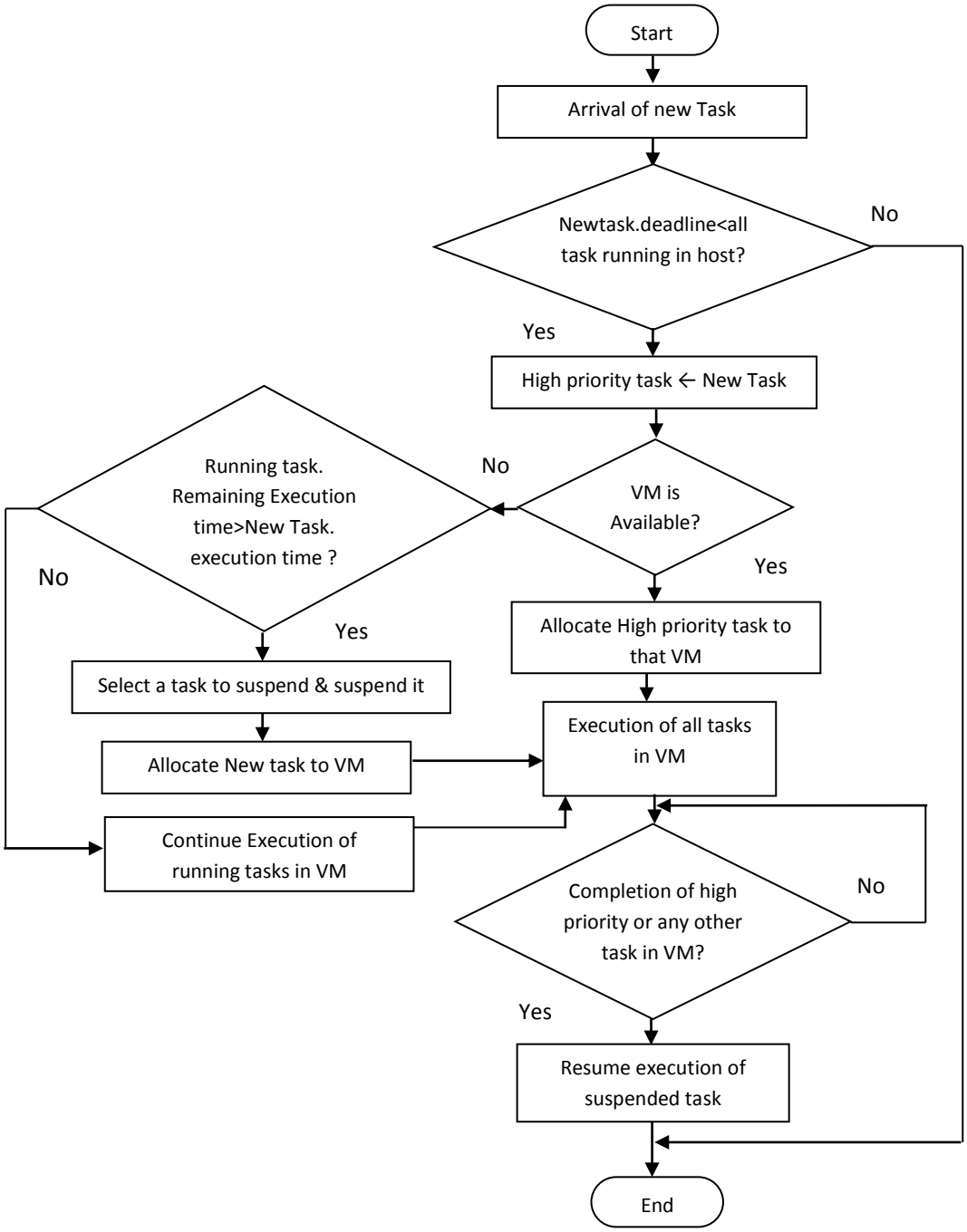

Figure 2. Flowchart of proposed algorithm 2

The proposed algorithms are only theoretically tested. In next phase of our work, we aim to implement our resource allocation algorithms .The outcome of experimentation will be compared with that of contemporary policies.

CONCLUSION

In this paper, we have identified the issue of handling priority task in Cloud environment with an aim to achieve higher resource utilization and minimized energy consumption. Various existing research papers have been surveyed and we have proposed a mechanism with modification in existing work of dynamic VM allocation based on task characteristics such as execution time and task deadline.

REFRENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Information Technology Laboratory 800-145, September 2011.

[2] Zahra Bagheri, Kamran Zamanifar, "Enhancing Energy Efficiency in Resource Allocation for Real-Time Cloud Services",2014/IEEE.

[3] Saraswathi AT, Kalaashri.Y.RA, Dr.S.Padmavathi, "Dynamic Resource Allocation Scheme in Cloud Computing",2015- Elseiver, doi: 10.1016/j.procs.2015.03.180

[4] S. S. Manvi and G. K. Shyam "Resource management for Infrastructure as a Service (IaaS) in Cloud computing: A survey," Journal of Network and Computer Applications, 41, ScienceDirect 2014, pp. 424-440.

[5] Chandrashekhar S. Pawar, Rajnikant B. Wagh, "Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue",2013International Conference on Intelligent Systems and Signal Processing (lSSP).

[6] Shubhakankshi Goutam, Arun Kumar Yadav, "Preemptable Priority Based Dynamic Resource Allocation in Cloud Computing with Fault Tolerance", 2015 IEEE International Conference on Communication Networks (ICCN)

[7] lksen Çalar, Deniz Turgay Altlar, "An Energy Efficient VM Allocation Approach for Data Centers", DOI 10.1109, 2016 IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, IEEE International Conference on Intelligent Data and Security.