I J C T A, 9(41) 2016, pp. 789-798 ISSN: 0974-5572

ENERGY EFFICIENT OPENSTACK VM SCHEDULE RUSING ANT COLONY OPTIMIZATION

G SudhaSadasivam1 and Raghuram D

Abstract: Ever increasing customer demands has resulted in an exponential raise in the number of physical machines in data centers. This trend has inevitably lead to huge electricity consumption. In Infrastructure-as-a-Service (IaaS) clouds, Virtual Machine (VM) placement is the process of mapping VMs to physical machines. An optimal VM placement improves power efficiency and resource utilization in a cloud computing environment. This paper suggests a multi-objective ant colony optimization algorithm for the VM placement problem. The goal is to efficiently obtain a set of hosts for VM placementto minimize resource wastage, power consumption and communication overhead. The proposed algorithm is implemented in OpenStack cloud environment. The performance of the proposed scheduler is compared with that of an existing filter scheduler.The results show that the proposed approach is more efficient than filter scheduler.

Key Words: *VM placement, Private cloud, Energy efficient scheduling, ACO*

I. INTRODUCTION

Cloud Computing[1] allows the user to access off-site, computing capabilities including storage and services over the Internet without having to maintain them. OpenStack [2] uses open source tools and technologies to deploy a private cloud. It is an Infrastructure as a Service (IaaS) cloud computing project.The OpenStack compute scheduler (nova) decides the host on which the VM is to be launched. It assigns the request to the node that has maximum resource to accommodate the process. The nova scheduler is not aware of the application to be loaded inVM. Further, it does not consider distance between nodes and energy efficiency. This leads to poor resource utilization and hence consumes enormous amount of energy. OpenStack compute scheduler's [3] VM assignment decision making does not depend on the CPU load; Rather hosts with large amount of CPU and memory are chosen, leading to resource wastage.

Conventionally a cloud scheduler considers QoS values, resource utilization, reliability for making scheduling decisions, without considering the communication bandwidth / distance between nodes. Distance parameter is of utmost importance in communication-intensive cloud applications. Dynamic resource allocation model[4] considers both the location of physical machines, and resource utilization parameters. Schedulers must consider distance between nodes in data center and also distance between the node and user.Location-aware applications can outperform those that are not location aware by factor. Hence, the network distance and bandwidth between the application and data could influence the performance of the application significantly [17]. Thus, it is necessary to control the location of the VMs so that the application hosted by the VM has a shorter data access time. To address this issue, a network aware VM placement and migration approach for data intensive applications in cloud computing

[1]*Department of Computer Science and Engineering, PSG College of Technology, Coimbatore,*
Email: sudhasadhasivam@yahoo.com
[2]*Wipro GE Healthcare pvt. Ltd, Bangalore, Email:* cberaghuram@gmail.com

environment is proposed. The proposed approach places the VMs on physical machines with consideration of the network conditions (mainly distance) between the physical machines and the data storage. IaaS instantiates VMs on Service Level Agreement (SLA) without considering the type of application loaded on VM. Based on the study of the effectiveness of Hadoop cluster [7] on AWS, a scheduler suitable for Hadoop in private cloud is designed. The parameters considered for making scheduling decisions include volume, compute capacity, CPU utilization and the distance between data nodes in the cluster.

II .LITERATURE SURVEY

The objective of a Green Cloud is to minimize energy consumption without compromising on performance by managing resources efficiently. A data center containing 1000 racks in 25,000 square feet requires 10 MW of power for the computing infrastructure per year and 5 MW for cooling. Energy management in cloud computing has become a challenge as number of data center is increasing. Energy efficiency can be achieved at hardware, operating system, virtualization and data center levels [1]. An energy-aware task consolidation (ETC)[3] technique has been proposed to increase resource utilization and minimize energy consumption.In Openstack.The work covers only computational resources; there is a need for including storage and networking resources.

Carbon emission based scheduling policy which considers energy efficiency factors such as energy cost, carbon emission rate, workload, and CPU power efficiency that vary based on the location, architectural design, and management of a data center can result in 30% of energy savings in comparison to profit based scheduling policies.e-STAB [12] reduces communication delay and congestion by effective traffic distribution. Compared to standalone mobile execution and cloud execution, the optimal collaborative execution strategy in mobile cloud environment [13] can significantly save the energy consumed on the mobile device. Energy efficient scheduling reduces energy consumption while still meeting SLA by slacking the non-critical jobs and allocating them in a global manner in the schedule [14].

Scheduling VMs that distribute maximum workload on minimum number of VM’s reduces energy consumption[15]. Higher CPU utilization usually implies greater energy consumption. Load management strategy for cloud [16] shifts the jobs on over-loaded physical machine to under-loaded physical machines. Energy Efficient VM Scheduling for Cloud Data Centers [17], presents optimal allocation algorithm based on bin packing problem with minimum power consumption. Ant Colony Optimization [18] can be used to solve bin backing problem.An improved clonal selection algorithm [19] with powerful global exploration capability based on time, cost and energy consumption models can be used for optimal resource allocation. A Multi-objective Optimization for VM placement in Cloud[20] uses Ant Colony Optimization (ACO) to achieve the optimal balance of the objectives resource wastage, power consumption and SLAviolation..Ant Colony Optimization (ACO) meta-heuristics is used to solve [21] VM consolidation in IaaS cloud.

Conventionally, the OpenStack compute (Nova) scheduler assigns the request to the node that has maximum resource to accommodate the process. But, the scheduler does not consider the application characteristics, distance between host nodes and energy efficiency. This leads to poor resource utilization and hence consumes enormous amount of energy. The paper uses multiple objectiveACO to improve VM placement considering power consumption, resource utilization and node proximity in Openstack cloud..

III. SYSTEM ARCHITECTURE

When user submits a request specification through schedule_run_instance method, the following activities are performed by the scheduler:

1) Selection of hosts: Hosts on which VMs are to be instantiated are selected as follows:
   a) Obtaining the list of all active hosts and the distance between the hosts
   b) Filtering and Ordering the hosts according to volume and compute requirements
   c) Obtaining the dynamic state of the hosts
   d) Passing the information to Ant Colony Optimization class
2) ACO: The following activities are carried out:
   a) Execution of ACO to evaluate resource utilization and power consumption at the hosts
      - Initializing ants, resetting pheromone, ant tours
      - Updating local pheromone based on distance, power consumption and utilization
      - Obtaining local and global best schedules
   b) Reordering and selecting the hosts for VM launch using global best schedule
   c) Passing the information to filter scheduler
3) VM instantiation: The VMs are instantiated by filter scheduler on suitable hosts.

The scheduler is deployed in Openstack controller node. The request by client is given to _schedule() method of AntScheduler class. This method will collect list of host states as unfiltered hosts.

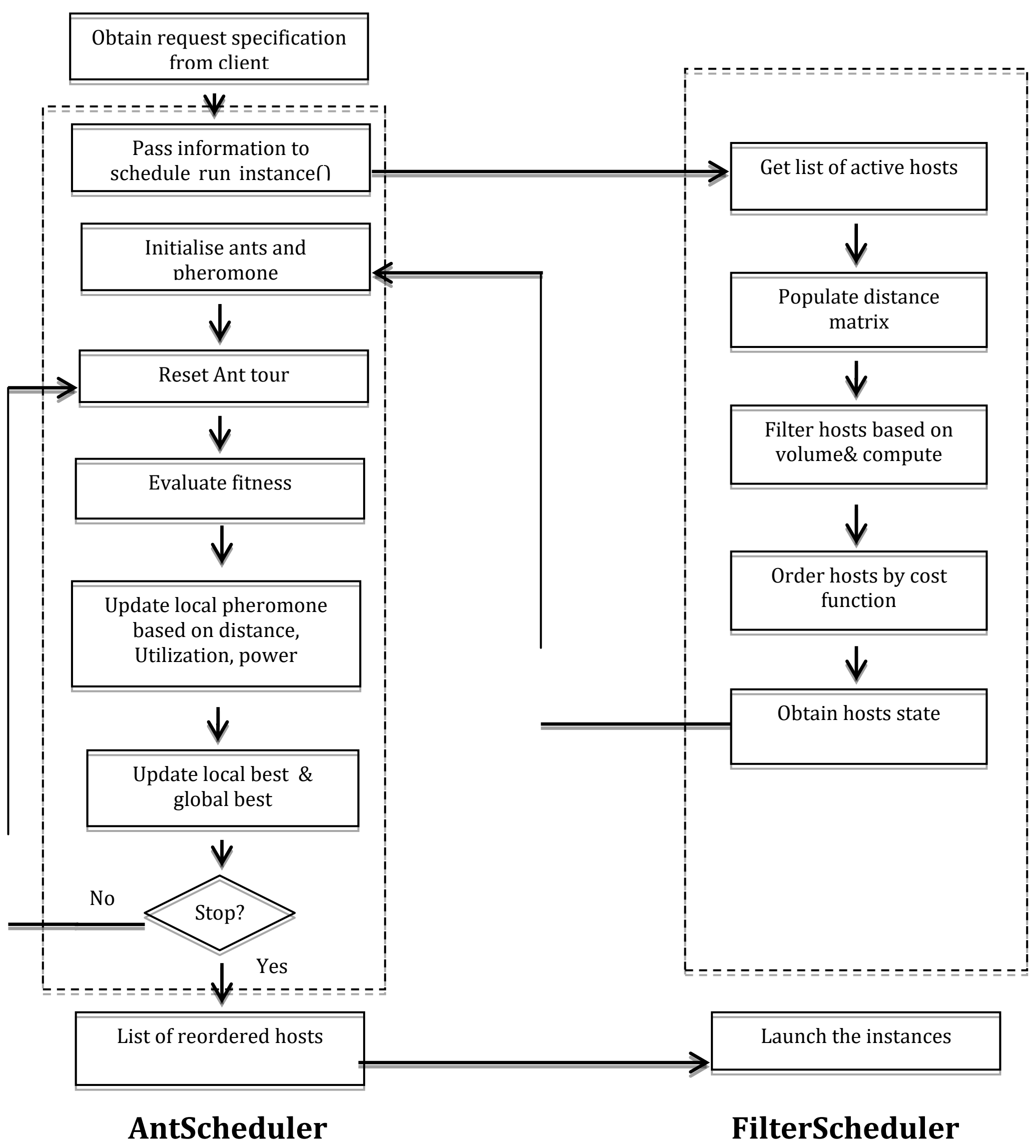


Figure 1.Architecture of the Ant Scheduler

Using request_spec and unfiltered_hosts as argument, a list of selected hosts is returned. AntColony class is used to execute ant colony optimization technique. Each ant initiates its tour and pheromone values are updated based on power consumption, distance and resource wastage. This process returns selected list of nodes to antcolony. Once the global value is updated, the final list of nodes is returned and VMs are launched in these nodes. The paper suggests ACO to find a near optimal solution.

3.1 ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a method for finding near optimal solution using a probabilistic technique. It can be used for problems which belong to the NP class, like the VM Placement Problem. Pheromones deposited by an ant controls behavior of other ants. Pheromone evaporation corresponds to the natural evaporation of chemical substance deposited by ants. It is used to reduce the amount of pheromone over time for keeping only trails that are regularly used. This system aim to promote trails which lead to better solutions. The following factors are to be considered for VM placement:

a) *RESOURCE WASTAGE:*In adistributed environment, balanced sharing of resources with minimum wastage is essential. To balance resource wastage, the potential cost of wasted main memory (RAM) among resources is calculated using Eq. (1).

$$RW = ((TR-UR) \div TR) * 100 \qquad (1)$$

Where, RW= Resource Wastage
TR = Total Resource
UR = Used Resource

The above equation relates the amount of resource used by the system with that wasted.

b) *POWER CONSUMPTION:* Power consumption by servers can be linearly related to CPU utilization. On average an idle server consumes approximately 70% of the power of a fully utilized server. Therefore, we define the power consumption as a function of the CPU utilization as shown in Eq. (2).

$$\text{Pj} = P_{idle} + \left(P_{busy} - P_{idle} * \text{Uj}\right) \quad (2)$$

$$\text{Uj} = \frac{\text{number of vcpus}}{\text{total number of vcpus}} \qquad (3)$$

Where Pj denote the power consumption and Uj denotes CPU utilization of the j^{th} server as calculated using Eq. (3). Total Power (P) is evaluated as $\sum_{j=1}^{m} P_j$ for all the selected servers (m). In our experiments, the values of P_{busy} and P_{idle} have been fixed at 69.1 and 185.8 Watts respectively.

Table 1. Categories of power consumption in Dell servers

% Utilisation (u)	Max power (W)	P_u
0-10	80.77	$P_0 + (P_{10} - P_0) * u$
11-20	101.77	$P_{10} + (P_{20} – P_{10}) * u$
21-50	143.78	$P_{50} + (P_{50} – P_{20}) * u$
51-100	185.8	$P_{100} + (P_{100} – P_{50}) *$

		u

c) *DISTANCE BETWEEN SERVERS*: Distance matrix to show the number of hops between the active servers is represented as follows:

$$\text{Distance Matrix} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1j} \\ d_{21} & d_{22} & \cdots & d_{2j} \\ \vdots & \vdots & \vdots & \vdots \\ d_{i1} & d_{i2} & \cdots & d_{ij} \end{pmatrix}$$

Where each d_{ij} indicates the distance between server i and server j. Total distance D $=\sum_{i,j=1}^{m} dij$ where m is the number of servers selected.

The multiple objectives to be balanced for VM placement is given as follows (where, m is the number of physical nodes, n is the number of VM requests):

$$\min \sum_{i,j=1}^{m} dij \text{ and } \min \sum_{j=1}^{m} Pj \text{ and } \min \sum_{j}^{m} RWj \quad (4)$$

The reciprocal of the objectives have to be considered to maximize the fitness function in ACO as given below:

$$\text{fitness} = \frac{1}{RW} + \frac{1}{P} + \frac{1}{D} \quad (5)$$

where P is power consumption, D is distance and RW is resource wastage

After all ants have finished building a solution, pheromone trails on all item-bin pairs need to be updated in order to help guiding the algorithm towards the optimal solution. Thereby, a pheromone trail update rule γ_{iu}(idenotes VM or item, u denotes host or bin) exists and is used in order to simulate pheromone evaporation and reinforce item-bin pairs which belonged to the so far best solution. In this work we follow the MAX-MIN Ant System (MMAS) approach in which only the iteration's-best ant (i.e., ant whose solution's objective function value is maximum) is allowed to deposit pheromone. The pheromone updated rule is defined as follow:

$$\gamma_{iu} = (1-\rho) * \gamma_{iu} + \Delta\gamma_{iu}^{best} \quad (6)$$

$$\Delta\gamma_{iu}^{best} = \begin{cases} f(S_{best}) & if\ a_{iu} = 1, \\ 0 & otherwise \end{cases} \quad (7)$$

where, the constant ρ, 0 < ρ<1 is used to simulate pheromone evaporation. Hence, higher values for ρ lead to increased evaporation rate. Moreover, some item-bin pairs need to be reinforced. Thereby, $\Delta\gamma_{iu}^{best}$ is defined as the iteration's-best item-bin pheromone amount. Hence, if some item belongs to a bin of the so far best solution S_{best}, its pheromone amount is reinforced. Consequently, only item-bin pairs which are parts of S_{best} will be reinforced and thus become more attractive. Others, which are not parts of S_{best} will continue loosing pheromone according to the pheromone evaporation rate ρ. **T**he probability for an ant k to choose an item I is chosen as the next one to pack in its current bin u:

$$P_{iu}^{k}(t) = \begin{cases} \dfrac{r_{iu}^{\alpha}(t) * \eta_{iu}^{\beta}(t)}{\sum_{s \in allowed_k} r_{su}^{\alpha}(t) * \eta_{su}^{\beta}(t)} & if\ i \in allowed_{k}, \\ 0 & otherwise \end{cases} \quad (8)$$

Whereby, γ_{iu} denotes the pheromone based desirability of packing item i into bin u , and k denotes VMs which the ant k have not searched or the host u can load. Moreover, two parameters α and β are used in order to either emphasize more the pheromone

The optimal solution of multi-objective optimization problem is usually a group of solution set. Exclusion method is used to construct the optimal solution set. Suppose that the solution set of an ant cycle is S_{cycle}= {s1, s2, ..., sl} (i is the number of the solution), a sub objective of the solution S_i is better than the solution S_j , and the other objectives of the solution S_i is not inferior than the solution S_j (i, j Є{1, 2, ..., l}), we can say S_i dominates S_j . Then the solution S_j can be deleted from the solution set S_{cycle}. This is repeated to get the optimalS_{cycle} . By repeating this procedure,theglobal optimal solution set S_{best} will be obtained.

The process of ACO for obtaining the optimal placement of VMs is listed as follows:

Step 1: Initialize all pheromone valuesand S_{best} = nil
Step2: Iterate to number of cycles
    Step 2.1: Iterate to number of ants
        Step 2.1.1: find the solution for each ant
        Step 2.1.2: Update the cycle best solution
    Step 2.2: Update the best solution
    Step 2.3: Update the pheromone trial
Step 3: return Best Solution

ANT SOLUTION:

Step 1: Initialize ant solution set, Item set, bins
Step 2: Iterate through number of items in item set
    Step 2.1: Select the capable nodes in to the bin
    Step 2.2: Calculate the probability using equation (8)
    Step 2.3: If the bin is selected for an item then update ant solution matrix

UPDATE PHEROMONE TRIAL:

Step 1: Iterate to number of items
Step 2: Update the pheromone value for all nodes using equation (6)
Step 3: If node is selected pheromone value will be more thannon selected nodes

IV. EXPERIMENTAL RESULTS

An openstack cluster was setup with 7 Dell servers in which one has all OpenStack controller components including nova-compute and the others have only the nova– compute component. Energy Star is given a Power and Performance Data Sheet for Dell PowerEdge R210 (250W Power Supply) which is considered by our scheduler. The distance matrix is formed that contains the number of hops between the available hosts. Each compute node information is stored as dictionary and added to list of compute nodes. We execute our optimization algorithm based on inputs. Algorithm returns list of selected nodes which is taken by OpenStack to instantiate VM on selected node. The algorithm is deployed in /nova/scheduler directory and configured it in nova.conf file

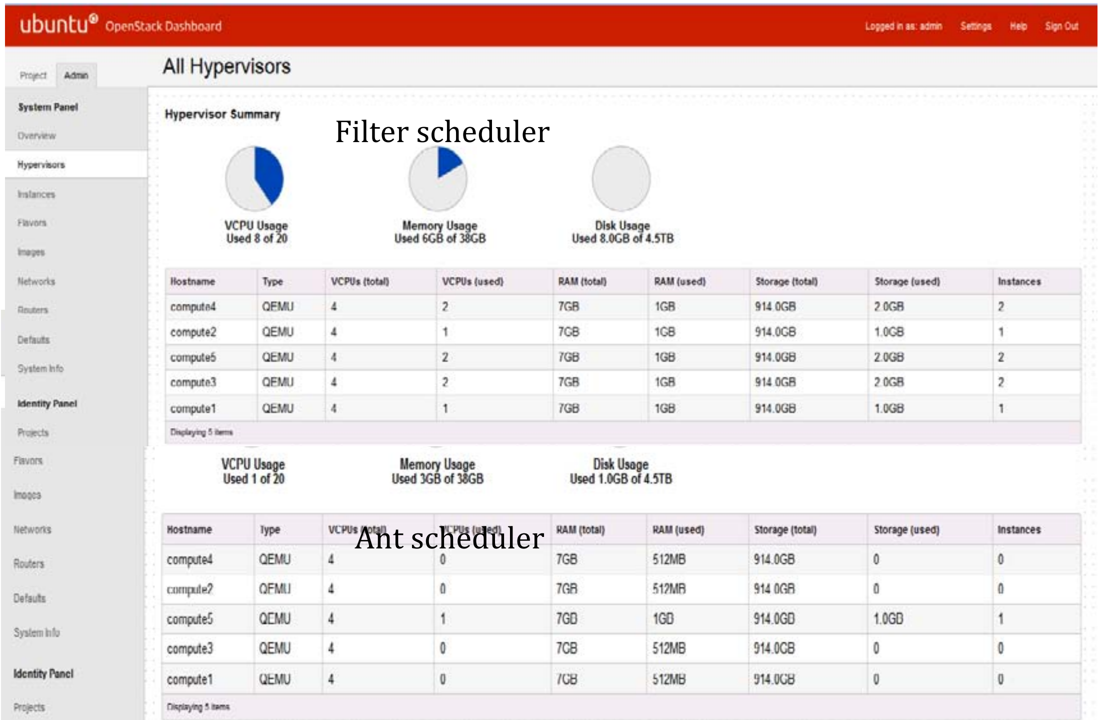


Figure 2: Request for 5 tiny instances in filter and ant scheduler

The Ant filter scheduler is implemented and its performance is compared with that of the existing filter scheduler in OpenStack. Users can create requests through the Dashboard and launch instances in hypervisor. If a user requests a 'tiny' type instance, the hosts are chosen randomly because all nodes have same amount of resource and instance is launched on selected node. In both filter and ant scheduler compute node 5 is chosen. If user request for four more instances simultaneously, filter scheduler chooses compute nodes 1 to 5 for five instances, whereas the ant scheduler chooses the same compute node 5 where the last instance was launched. Thus, Filter scheduler of OpenStack schedules instances horizontally to utilize all the servers. Ant scheduler creates instance on same nodes until resource can be accommodated. Eight tiny instances are launched in compute node 5 for ant scheduler. Nodes 1 to 4 are in idle state. Filter scheduler launches 2 instances in compute nodes 3, 4,5 and one instance in nodes 1,2.

Memory wastage and Power consumption of the two schedulers is calculatedand is tabulated in table 2. Ant scheduler can accommodate a request with 7 GB memory whereas filter scheduler cannot.

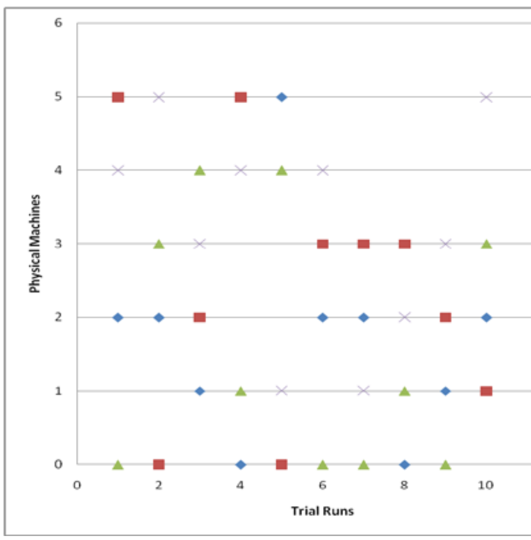
Table 2 – Energy / Memory consumed by servers

Scheduler	Power used by servers(Watts) / Memory available in GB										Total Power Used(Watts)
	compute 1		compute 2		compute 3		compute 4		compute 5		
	Watts	GB	Watts	GB	Watts	GB	Watts	GB	Watts	GB	
Filter	122.7	6.5	22.7	6.5	143.7	6	143.7	6	143.7	6	676.5
Ant	69.1	7	69.1	7	69.1	7	69.1	7	185.8	3	462.2
Total Energy Saved is 214.3 W											

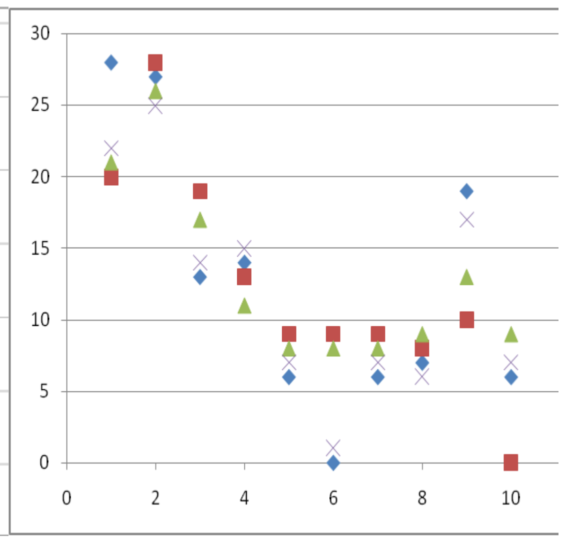
Consider the following scenario with 6 hosts in the cloud environment

Host id	VCPU	VCPU used	RAM (MB)	RAM used	Disk (GB)	Disk used
Host 0	4	3	1024	768	1000	750
Host 1	4	2	1024	512	1000	500
Host 2	4	2	1024	512	1000	500
Host 3	4	1	1024	256	1000	250
Host 4	4	0	1024	0	1000	0
Host 5	4	0	1024	0	1000	0

If an instance is requested with 256 MB RAM, 5 GB hard disk and 1 VCPU, the ant scheduler has allocated the instance on different hosts during the various trail runs as shown in the figure 3(a). The frequently selected physical machine to launch the VM is node 2. This is followed by host 1, host 0 and host 5. To verify, the power savings and resource savings are evaluated and tabulated in table 3.



(a)



(b)

Figure 3.(a) Node selection for scenario 1 (b) Node selection for scenario 2

Table 3. Evaluation of Scenario 1

Scheduler	% Select	Node Selected	Power Consumed by host before /after VM launch (W)		RAM wastage(MB)	Data center before /after VM launch (W)		Additional power usage (W)
			Before	After		Before	After	
Filter scheduler	50	4	69.1	122.7	768	726.59	780.19	53.6

Filter scheduler	50	5	69.1	122.7	768	726.59	780.19	53.6
Ant Scheduler	50	2	143.7	178.29	256	726.59	761.18	34.59
Ant Scheduler	20	1	143.7	178.29	256	726.59	761.18	34.59
Ant Scheduler	20	0	178.29	185.8	0	726.59	734.1	7.51
Ant Scheduler	10	5	178.29	185.8	768	726.59	780.19	53.6

For another scenario (figure 3 (b)) with 30 nodes in 3 racks, nodes 0-9 are placed in rack 1, 10-19 in rack 2 and 20-29 in rack 3. Distances between racks are also considered for node selection. Ant scheduler chooses set of nodes inside the same rack (hosts 9, 8 and 6) for each trial run. Increase in power and wastage of resources is also reduced.

V. CONCLUSION

In cloud environment, resourcesare distributed, heterogeneous and virtualized. As a cloud computing system aims at maximization of profit, the amount of energy consumed by these systems has a big influence on profitability. Much of the literature shows energy consumption and resource utilization in clouds are highly coupled. Design and implementation of an ant scheduler for OpenStack cloud environment to minimize energy consumption and maximizing resource utilization is presented. As it places VMs on nodes in close proximity, communication intensive application can be deployed. The ant scheduler reduces the power usage, memory wastage, improves resource utilization and also reduces communication distance between nodes.

REFERENCES

[1] Cloud Computing Concepts, Technology & Architecture Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, pearson, 2013.

[2] Introduction to OpenStack and its components, http://docs.openstack.org/diablo/openstackcompute/starter/content/Introduction_to_OpenStack_and_its_components-d1e59.html.

[3] YongShengGong, “OpenStack nova-scheduler and its algorithm”, IBM Developers work, 2012.

[4] Fetahi Wuhib, Rolf Stadler and Hans Lindgren , “Dynamic Resource Allocation with Management Objectives-Implementation for an OpenStack Cloud”, KTH TECHNICAL REPORT, 2012.

[5] Vikram Yadav, G. Sahoo and K. Mukherjee, “Cluster Energy Optimization: An Algorithmic Approach”, May 2013.

[6] Jing Tai Piao and Jun Yan, “A Network-aware Virtual Machine Placement and Migration Approach in Cloud Computing”, IEEE International conference on grid and cloud computing, 2010.

[7] Gayathiri N., Sudha Sadasivam G,“Evaluation of hadoop on local clusters and amazon public cloud”, International conference on cloud and big data analytics, India (2013).

[8] Raghuram D, Gayathiri.N, Sudha Sadasivam.G, "Efficiently scheduling hadoop cluster in cloud environment", Proceedings of International Conference on Informatics and Communication Technologies for Societal Development (ICICTS14).

[9] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, and Albert Zomaya, "A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems", Technical Report, Cloud Computing and Distributed Systems Laboratory, 2010.

[10] Antonio Corradi, Mario Fanelli and Luca Foschin, "VM consolidation: A real case based on OpenStack Cloud", Future Generation Computer Systems, Elsevier, 2013.

[11] Saurabh Kumar Garg a, Chee Shin Yeob, Arun Anandasivamc, Rajkumar Buyyaa J, Parallel Distrib. Comput. 71 (2011) 732–749.

[12] Dzmitry Kliazovich, Sisay T. Arzo, Fabrizio Granelli, Pascal Bouvry and Samee Ullah Khan,"e-STAB: Energy-Efficient Scheduling for Cloud Computing Applications with Traffic Load Balancing".

[13] Weiwen Zhang , Yonggang Wen , Wu, D.O.,"Energy-efficient scheduling policy for collaborative execution in mobile cloud computing".

[14] Christoph Aschberger, Franziska Halbrainer,"Energy Efficiency in Cloud Computing", July 18, 2013 IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.

[15] Shailesh S. Deore and Ashok Narayan Patil, "Energy-Efficient Scheduling Scheme for Virtual Machines in Cloud Computing", International Journal of Computer Applications, October 2012.

[16] Ching-Hsien, Kenn D. Slagter, Shih-Chang Chen and Yeh-Ching Chung, "Optimizing energy consumption with task consolidation in clouds", Elsevier: Information Sciences, 2012.

[17] Saurabh Kumar Garg, Chee Shin Yeob, Arun Anandasivam C, Rajkummar Buyyaa J, "Environment-conscious scheduling of HPC applications on distributedCloud-oriented data centers", Parallel Distribution computing 71 (2011) 732-749.

[18] Marco Dorigo and Thomas Stutzle, "Ant Colony Optimization", the MIT Press, 2004.

[19] Wanneng Shu**,** Wei Wang **and** Yunji Wang, "A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing", *EURASIP Journal on Wireless Communications and Networking* 2014, **2014**:64.

[20] Fei MA, Feng LIU and Zhen LIU, "Multi-objective Optimization for Initial Virtual Machine Placement in Cloud Data Center", Journal of Information & Computational Science, 2012.

[21] Armel Esnault, "Energy-Aware Distributed Ant Colony Based Virtual Machine Consolidation in IaaS Clouds", white paper, 2012.