

# Cluster Energy Optimization: An Algorithmic Approach

Vikram Yadav  
Department of Computer  
Science & Engineering  
Birla Institute of Technology  
Mesra, Ranchi, India

G. Sahoo  
Department of Information  
Technology  
Birla Institute of Technology  
Mesra, Ranchi, India

K. Mukherjee  
Department of Computer  
Science & Engineering  
Birla Institute of Technology  
Mesra, Ranchi, India

## ABSTRACT

In fact, Gartner projected global revenue for cloud computing to reach almost \$150 billion by 2014. However, The 2011 market is already approx \$68 billion globally. With increase in web technologies and Internet, a proportional increase in Cloud computing technologies has been cited. Cloud computing has been emerging as a flexible and powerful computational architecture to offer ubiquitous services to users. A variety of hardware and software resources are integrated together as a resource pool, the software is no longer resided in a single hardware environment, it is performed upon the schedule of the resource pool for optimized resource utilization. The optimization of energy consumption in the cloud computing environment is the question how to use various energy conservation strategies to efficiently allocate resources. The need of different resources in cloud environment is unpredictable. It is observed that load management in cloud is utmost needed in order to provide QOS. The jobs at over-loaded physical machine are shifted to under-loaded physical machine and turning the idle machine off in order to provide green cloud. For energy optimization, DVFS and Power-Nap are good strategies. As much of this energy is wasted in idle systems: in typical deployments, server utilization is below 30%, but idle servers still consume 60% of their peak power draw. In this paper, we have proposed an algorithm for energy optimization having the constraint QOS and SLA.

## Keywords

Cloud Computing, DVFS, Power-Nap, Ant colony algorithm, bee colony algorithm etc.

## 1. INTRODUCTION

Google and Apache foundation launched GFS, Big-Table, MapReduce [1] [2] [3] and Hadoop [4] respectively. Hadoop is a free java based framework that support the processing of large data sets in a distributed computing environment. Hadoop use the concept of google's MapReduce. Amazon launched simple storage services(S3) [5] in 2006. It is a scalable, low cost, high speed web based service designed for online backup and achieving of data and application program. Datacenter are estimated to consumed 2% of the U.S. electrical grid capacity resulting \$2 billion per monthly in utility bills. A McKinery & company study estimates carbon dioxide emission from datacenters will quadruples to exceed emission from the airline industry by 2020 owing to rapid growth in global demand for computing power [6]. Unfortunately, Maximum of the energy is wasted by systems that are idle. At idle, current servers still draw about 60% of

the peak power [7-9]. In typical data-centers, average utilization is only 20-30% [7] [10]. With the variety of cloud computing model and their framework, some of the computing nodes and data centers form a cluster. Google has many datacenters, one of them in Oregon consumes equal to a medium size city [11]. During the peak operation, a total of 29.4MW of electricity is consumed by Dublin Datacenter [12]. Idle servers or under-loaded servers impact double on the performance.

Organization of this paper is as follows: Related Work is discussed in section 2. Existing framework of Eco-Friendly Cloud Computing is discussed in section 3. Proposed methodology is discussed in section 4. Proposed algorithm is discussed in section 5. Section 6 gives details of the experimental results. Section 7 gives the conclusion with a direction of the future work.

## 2. RELATED WORK

David Meisner [13] presented a power nap strategy for eliminating server idle power. Liang Luo [14] presented a resource scheduling algorithm based on energy optimization methods. Eugen Feller [15] summarized the energy consumption and adjustment methods of standalone computer and cluster servers. In Chung-Hsing Hsu's work [16], They continuously observed the energy consumption and server utilization of server from 2007 to 2010, pointing out that server's defaults energy model has changed. In Charles lively & Xingtu Wu's work [17], they use precise experimental probes to collect energy consumption and performance characteristics in different parallel implementations. HoAydin et.al [18] have proposed the minimizing the energy consumption and subsequently the cost for the static system. K.Mukherjee & G. Sahoo [19] have proposed a framework for achieving better load balancing in Grid environment. Again, K. Mukherjee & G. Sahoo [20] proposed an ant colony optimization and Bee colony for service rescheduling. Once again, K. Mukherjee & G. Sahoo [21] proposed an algorithmic approach for Green Cloud.

## 3. ENERGY CONSUMPTION OF CLOUD CLUSTERS

### 3.1 STRUCTURE OF CLUSTER

In order to design a cluster energy optimization technique, we must know the components. A cluster node is mainly composed of three node – controller node, compute node and storage node. A compute node is collection of more than one

multi-core CPU's, controller node is a collection of switching nodes and storage node is a collection of multi-channel memory.

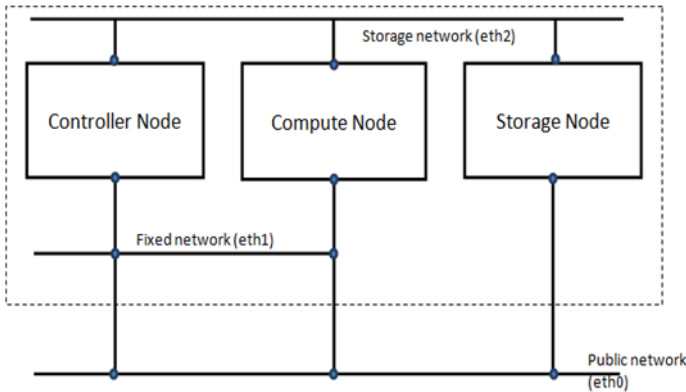


Fig . 1 General structure of Cloud cluster node

### 3.2 ENERGY MODEL OF CLOUD ENERGY ENVIRONMENT

The overall energy consumption of a cloud computing system can be expressed as the following formula:

$$E_{cloud} = E_{controllernode} + E_{computenode} + E_{storagenode}$$

$E_{storagenode}$  represents the energy consumption by multi-channel memory,  $E_{computenode}$  represents the energy consumption by the computing nodes and  $E_{Controllernode}$  represents the energy consumption by the controlling nodes.

## 4. PROPOSED METHODOLOGY

We have proposed two algorithms for different purposes.

- (1) Ant Colony optimization for energy optimization.
- (2) Bee Colony optimization for service rescheduling.

### 4.1 ANT COLONY OPTIMIZATION

Ant Colony optimization technique was proposed by Marco Dorigo in the early of 90's. In our problem, there are  $n$  nodes in a cluster. Some are under-loaded, over-loaded and idle. Ant colony optimization will help in traversing all nodes by shortest path (Traveling salesman problem). After traversing all the nodes, we will find the nodes that are idle and make them turn off. In our practical implementation, we have provided a color code (RGB code) to each node. We will set a Threshold value like summation of RGB must be less than or equal to 300. The node will be turn off or on on the basis of RGB scale. Ants (blind) navigate from source to food with the help of Pheromone. Initially, the ants navigate in all possible paths from source to food uniformly. On the long path, ants take more time to return as there will be more distance between the ants. So, the evaporation will be high as compared to small path, there will be less distance between the ants and the evaporation will be less. In the next iteration, more ants will navigate in that path that has more pheromone means shortest path. So, shortest path is discovered via pheromone trails. Each ant move in random, Pheromone is deposited on path. Ants detect the lead ants path inclined to follow more pheromone path. The probability of path being followed starting node are selected at random path selected at

random based on amount of trail present on possible paths from starting node. The path that has higher probability has more trails. Ant reached next node, select path and continues until reaches start node. Finished tour is a solution. A completed tour is analyzed for optimality. Trail amount adjusted for favor better solution. A better solution receive more trails. A worse solution receive less trail. Higher probability of ant selecting path that is a part of better solution. A better solution receive more trail. A worse solution receive less trail. Higher the probability of ant selecting path that is a part of a better performing tour. New cycle is performed repeated until most ants select the same tour on every cycle.

### 4.2 BEE COLONY OPTIMIZATION

Bee Colony optimization is used for service rescheduling. There are many nodes that are under-loaded and some are overloaded. We will uniformly distribute the load among all the node. Honey bee is a social insect. They work in a decentralized and well organized manner [20]. There are two types of bees: One is forger bees, who collect the nectar and food-stores, who store that in hives. Forger bees moves out for searching the nectar(food). They move randomly in any direction. After finding the nectar, they come back on the hive and start dancing on the dance floor. The duration of this dance is closely correlated with the search time experienced by the dancing bees. There are two types of dance, waggle dance which implies poor quality of nectar and tremble dance (round dance) which implies good quality of nectar. If the dance is tremble dance, then the new born bee agents fly to collect the good quality nectar and store them in the hive. After this operation, the old bee agents die and the new born bee agents start to fly with the good quality nectar stored in the hive, and finally mix them with those sources which are holding poor quality nectar. This process of distribution goes on until there is a uniform quality of nectar in all the sources. Similarly, in cloud computing environment, we observe that some CPUs of IaaS are overloaded for processing consumers services, some are under loaded and some are totally idle. We can save the consumption of energy by turning this idle CPUs OFF and rescheduling services from overloaded CPUs to under loaded CPUs.

### 4.3 HYBRID APPROACH

For this the combined Bee-Ants colony system can be used. Firstly, we can divide our jobs into two parts, the first part, which looks after the proper management of overloaded and under loaded CPUs (service rescheduling) and the second part, which helps to manage the idle CPUs. We propose bee colony algorithm for service rescheduling and ant colony algorithm for power consumption management. Thus, for service rescheduling, we can think the tracker as hive which consist of bee agents, from where agents start to forge for nectar. Here nectar implies the threshold value of CPU. Poor quality of nectar implies lower threshold value of CPU whereas good quality of nectar implies the higher threshold value of CPU. Good quality of nectar implies overloaded CPU and poor quality of nectar implies under loaded CPU, in addition the dance floor represents the service scheduling table, where the information about the status of the CPUs that the bee agents have visited are stored. Actually this

process is implemented for rescheduling the services from overloaded CPU's to under loaded CPU's in order to provide good QoS to consumers. Again, if the dance of the bee agents is waggle dance then hive tracker indicates the service acceptor to accept new services from consumer. After this step, we apply ant colony based algorithm to find the idle CPUs and then turning them OFF in order to minimize the consumption of power and hence, lower the operational cost. The ant colony from VM turns ON-OFF decider start to find the idle CPUs. The algorithm involves artificial agents called ants who show a cooperative behavior to find the shortest path to the food source from their nest. Ant System is best suited in solving problems like travelling salesman problem which run in dynamic and varied environments without the help of any central control. This is why it is very much applicable to distributed problem solving. As cloud computing is dynamic in nature, hence we propose to apply ant system in cloud computing in finding idle CPU's. The self-organized behavior of this social insects can be used to solve problems in cloud computing. Here the ants are analogous to cloud data centers and food foraging to load allocation to reduce power consumption. We can make analogy with ant colony algorithm for the cloud environment considering its dynamic nature. The creation of controller i.e. Queen ant divides the VMs into two parts: the servers those are in ON state and the servers those are in OFF state. Worker ants are comparable with cloud servers, queen ant with randomly selected cloud server, its reproduction means switching more servers in ON state when needed and cleaner ant's dead body removal means switching OFF the unwanted or idle servers to reduce power consumption. The main challenges in this kind of environment is to manage the state of cloud servers ON and OFF by the controller. When the allocation is already done by using bee colony algorithm, the service scheduling table is updated containing the attributes like remaining power etc. which is maintained by the controller itself (queen ant) and is updated time to time which helps to make decisions that which servers should be ON or put into sleep mode as they are idle.

## 5. PROPOSED ALGORITHM

At the first, the consumer requests for service to Graphical user interface for Consumer(GUIC). From here the request for service is transferred to Service Acceptor(SA).SA decide on the basis of Tracker of Virtual machine(TVM) that service should be accepted or rejected. If service is accepted, check for Quality of service(QoS) checker. After that, services are submitted to execution manager which interacts with power consumption policy maker(PCPM). PCPM interacts with virtual machine turn on-off decider(VMTD). Then, it assigns services to VMs. A record tracker tracks the record. Active VMs execute the services on CPUs of IaaS. It turns the powers of the corresponding CPUs of IaaS ON and the rest are made OFF or idle. Then, it sends the services from active VMs to active CPUs of IaaS using mapping of service from virtual machine to IaaS. Thus, all CPUs of IaaS are not turned on, only those required are turned ON and the rest are turned OFF. In this way power consumption of energy is minimized along with minimization of operational cost and the emission of carbon dioxide gas to the environment. Thus cloud computing can be a very promising technology due to its eco-friendliness and minimized operational cost.

The following algorithm is proposed here for service rescheduling (bee colony optimization) and energy management (ant colony optimization).

*/\* bee colony optimization\*/*

Consumer request to GUIC for service and GUIC submit request to SA which checks virtual machine on TVM.

If (Virtual machine are available)

{

It will check PCPM

If(utilization power is not within the acquired range)

Required service can't be fulfilled

else (check for service violation and cost)

Required service can be fulfilled

If(virtual machine not available)

Request for service is rejected.

*/\* ant colony optimization\*/*

Initialize Trail

Do while (stopping criteria Not Satisfied)-cycle loop

Do until (Each Ant complete a tour)- Tour loop

Local trail update

End do

Analyze tours

Global trail update

End Do

The algorithm for the above is given:

Virtual\_machine\_turn\_ON/OFF\_decider()

Begin

{

$\tau_{rs} = \tau_0;$

$\eta_{rs} = \eta_0;$

$H_r = H_0;$

}

Place n ants at the nest repeat till the final path is decided.

For every edge(r,s) do

$\tau_{rs}(0) = \tau_0;$

$$\eta_{rs} = \eta_0 ;$$

For t=1 to  $t_{max}$  do

For k=1 to m do

Build tour  $T^k(t)$

Choose the next CPU with probability

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)] \cdot [\eta(r,s)]^\beta \cdot p(s)}{\sum_{u \in J_k(r)} [\tau(r,u)] \cdot [\eta(r,u)]^\beta} & \text{if } s \in J_k(r) \\ 0, & \text{otherwise.} \end{cases}$$

Find the threshold  $H_s$  of  $s^{th}$  CPU.

$$p(s) = \begin{cases} 1 & \text{turnoff, if } RGB(H_s) < RGB(H) \\ < 1 & \text{move to next node} \end{cases}$$

where :

m = the number of ants.

$\tau(r, s)$  = Pheromone between node r and s.

$\rho$  = pheromone evaporation rate.

$\eta_{rs} = 1/d_{rs}$ , called visibility

$d_{rs}$  = distance between node r and s nodes.

$\beta$  = visibility deciding parameter

$J_k(r)$  = set of all neighbours of node r

$H_s$  = RGB Color code of  $s^{th}$  CPU.

$p(s)$  = probability of  $s^{th}$  node.

We define the following Global updating rules:

Once all ants move built their complete tours , pheromone is updated on all edges as follows:

$$\tau((r,s) \leftarrow (1 - \rho) \cdot \tau(r,s) + \sum_{k=1}^m \Delta\tau_k(r,s)$$

Where

$$\Delta\tau_k(r,s) = \begin{cases} \frac{1}{L_k} & \text{if } (r,s) \in \text{tour done by ant } k. \\ 0 & \text{otherwise,} \end{cases}$$

Where:

$L_k$  = sum of lengths of arcs.

## 6. EXPERIMENTAL RESULTS

A simulation for the ant colony optimization has been set up on a dual-core CPU, 512MB RAM, 120 GB hard-disk. In this Simulation, we have assigned a number of nodes having different RGB color. We assumed whose RGB color code summation is below, we have to switch off that node. On the last page, we have seen that simulator shows the number of ants, ants speed, pheromone evaporation rate and Pheromone quantity. It will turn off those nodes that have minimum RGB code summation.

Initially, we have to traverse all the node in minimum time interval. For that we have so many algorithm but ant colony system holds good. As, we can see from the results given below that ACS traverse minimum distance to perform all node visit.

The heuristics with which we compare ACS in this case are a genetic algorithm (GA), evolutionary programming (EP), and simulated annealing (SA). ACS is run for 1,250 iterations using 20 ants (this amounts to approximately the same number of tours searched by the heuristics with which we compare our results). ACS results are averaged over 15 trials. The distances are taken in floating points. This practical has been performed on Simulator.

Proble m name	ACS	SA	EN	SO M	FI
Node set 1	<b>5.86</b>	5.88	5.98	6.06	6.03
Node set 2	<b>6.05</b>	6.01	6.03	6.25	6.28
Node set 3	<b>5.57</b>	5.65	5.70	5.83	5.85
Node set 4	<b>5.70</b>	5.81	5.86	5.87	5.96

Node set 5	6.17	6.33	6.49	6.70	6.71
------------	------	------	------	------	------

Table1.Comparisons on average tour length obtained on five 50-nodes problems.

**Discussion:** In the given result, we have taken five different samples with 50-nodes. The length of path being traversed in the traversal of all the nodes is calculated. We can see the length of path through Ant Colony System is nearly same with simulated annealing but this difference increases as the number of nodes increases. As, we can see that FI gives worse result in all the five algorithms. So, we can say that ACS is better than other algorithms proposed for Traveling salesman problem.

## 7.CONCLUSIONS WITH FUTURE DIRECTION OF WORK

In this paper we have emphasized on developing an eco-friendly cloud computing by turning the switches of idle CPUs of IaaS off, simultaneously we have focused on avoiding SLA violation for minimized energy consumption and operational cost. For the overall process, we have proposed algorithm. We feel that a hybrid approach to be used that consists of Power-Nap, DVFS and ant colony optimization. So that every techniques may gives its better part. As Security and QOS are constraint in Cloud computing, they should be prevented.

## REFERENCES

- [1] The Google File System, <http://labs.google.com/papers/gfs-sosp2003.pdf>.
- [2] Bigtable:A Distributed Storage System for Structured Data , <http://labs.google.com/papers/bigtable-osdi06.pdf>
- [3] MapReduce:Simplified Data Processing on LargeClusters, <http://labs.google.com/papers/mapreduceosdi04.pdf>
- [4] Hadoop, <http://lucene.apache.org/hadoop/>
- [5] Amazon Simple Storage Service, <http://aws.amazon.com/s3/>.
- [6] The Datacenter Journal, <http://www.datacenterjournal.com/facilities/the-green-data-center-opportunity/>.
- [7] L. Barroso and U. Holzle, "The case for energy-proportional computing,"IEEE Computer, Jan 2007.
- [8] X. Fan, W.D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in Proc. of the 34th Annual InternationalSymposium on Computer Architecture, 2007.
- [9] C. Lefurgy, X. Wang, and M. Ware, "Server-level powercontrol,"in Proc. of the IEEE International Conference on Autonomic Computing, Jan 2007.
- [10] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, and R. Rajamony, "The case for power management in web servers,"Power Aware Computing, Jan 2002.
- [11] <http://www.netxt.com/power-103-megawatt-secret-google-container-data-center/>
- [12] Microsoft Dublin Data Center, <http://www.datacenterknowledge.com/inside-microsofts-dublin-mega-data-center/dublin-data-center-generators/>
- [13] D.Meisner and B.Gold, "PowerNap: Eliminating Server Idle Power", ACM 978-1-60558-215-3/09/03.
- [14] Liang Luo, Wenjun Wu "A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods" IEEE-2010.
- [15] E.Feller., D. Leprince and C. Morin. "State of the art of power saving in clusters results from the EDF", case study. 2010.
- [16] C.H.Hsu & S. W. Poole. "Power Signature Analysis of the SPECpower\_ss2008 Benchmark[C]. Performance Analysis of Systems and Software (ISPASS)", 2011 IEEE International Symposium, 2011: 227-236.
- [17] C.Lively , X. Wu, V. Taylor, S. Moore, H. Chang and K. Cameron. "Energy and performance characteristics of different parallel implementations of scientific applications on multicore systems", International Journal of High Performance Computing Applications, 2011, 25(3): 342-350.
- [18] H. Aydin, R.G Melhem, D Mosse, and P. Mejia-Alvarez. "Power-Aware Scheduling for Periodic Real-Time Task", IEEE Transactions on Computers, May 2004, , 53(5)
- [19] K.Mukherjee and G.Sahoo "Mathematical Model of Cloud computing framework using Fuzzy Bee Colony optimization Technique", International Conference on Advances in Computing, Control and Telecommunication Technologies, IEEE Xplorer, 2009.
- [20] K.Mukherjee and G.Sahoo, "A framework for achieving better load balancing and job scheduling in Grid environment", International Journal of Information Technology and knowledge Management, volume 2, No - 1, pp- 199- 202, January- June, 2009.
- [21] K.Mukherjee. and G.Sahoo. "Green Cloud: An Algorithmic Approach", International Journal of Computer Application Volume9 – No.9 November 2010.

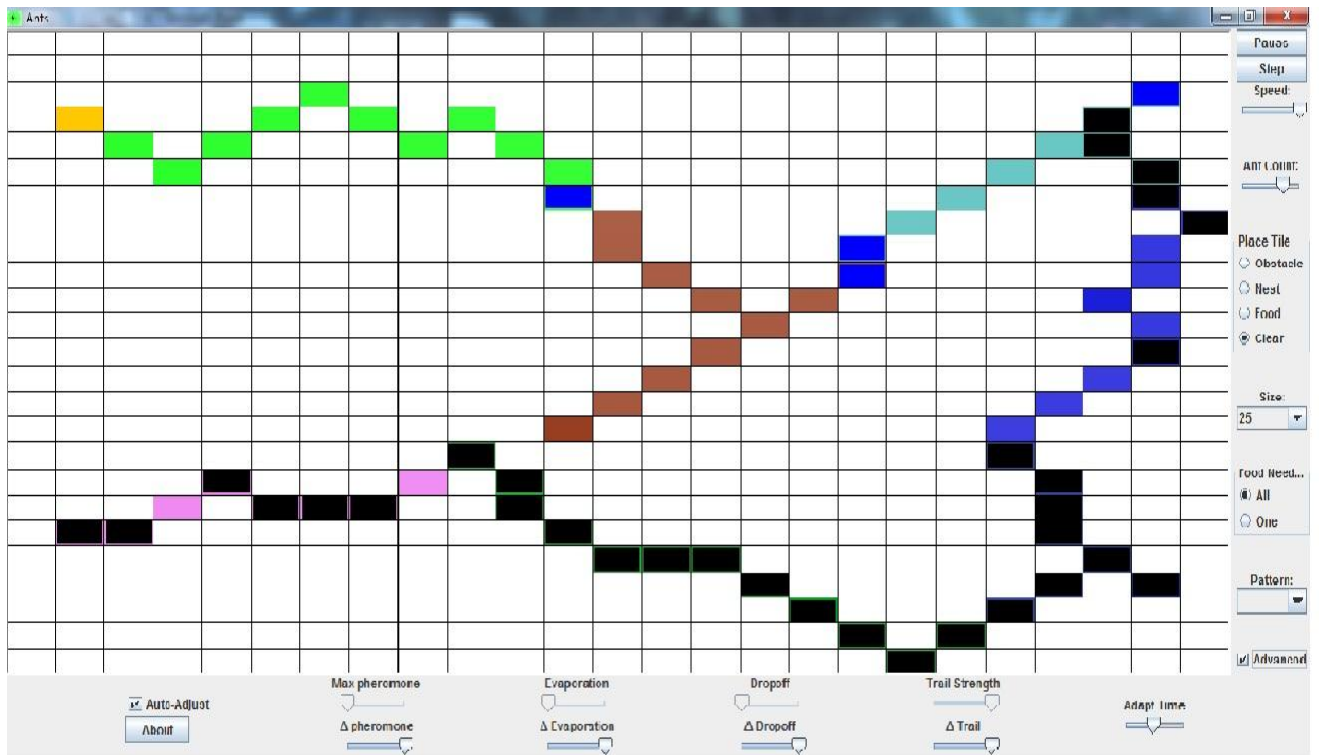


Fig.2 Ant Colony simulator base on Travelling Salesman Problem