# Improving Energy Consumption in Cloud

Sambit Kumar Mishra, Reenu Deswal, Sampa Sahoo, Bibhudatta Sahoo
Department of Computer Science and Engineering
National Institute of Technology, Rourkela, India
Email: (skmishra.nitrkl, reenu.deswal29, sampaa2004, bibhudatta.sahoo)@gmail.com

***Abstract*—To meet the service level agreement (SLA) between the cloud user and the cloud service provider, the service provider has to pay more. The cloud resources are allocated not only to satisfy the quality of services (QoS) those are specified in SLA, but also need to reduce energy utilization. Therefore, task consolidation plays an important role in cloud computing, which map users service requests to appropriate resources resulting in proper utilization of various cloud resources. The enhancement of overall performance of cloud computing also depends on the Task Consolidation approaches. Here, for task consolidation problem, we present an energy aware model which includes description of physical hosts, virtual machines and service requests (tasks) submitted by users. For the proposed model, an Energy Aware Task Consolidation (EATC) algorithm is developed where heterogeneity also affects the performance and show significant improvement in energy savings.**

***Keywords*—** Cloud Computing, VM, QoS, Energy.

## I. Introduction

With the recent advancements going in the field of computer technology, Cloud computing has emerged as an important paradigm that provides scalable and dynamic virtual resources to the users on demand through the internet. Cloud computing is a delivery model that delivered the on-demand computing resources from applications to data center over the Internet on a pay-for-use basis. According to the researchers in [5], [11], [14], [15], the cloud computing is a model which enables convenient and on-demand network access to a shared pool of computing resources (e.g., networks, servers, storage, applications, and services), that will be done with minimal management effort.

The main advantage of cloud environment is that it reduces the hardware cost and users can access high quality services at a low cost. Large Internet companies like Google and Microsoft have significantly improved the energy efficiency of their multi-megawatt data center, focusing mostly on hardware aspects. According to [2], the energy consumed by computers was around 2% of the total electricity consumption in the US. One of the major reasons of energy inefficiency is idle power wastage. Even at very low load, i.e. around 10%, the power consumed is 50-60% of the peak power [16], [17], [18]. The Microsoft Dublin Data Center will consume 5.4 MW of electricity and may be expanded to 22.2 MW in the near future [4]. The Tianhe-1, a cluster computer in Tianjin, China, consumes 128KW electricity per hour. This is equivalent to the electricity consumption of 2 million ordinary families [4].

The problem of energy efficient allocation of different virtualized resources (processor, database servers, RAM, network, etc.) is complex because of the heterogeneous nature of workload application having different resource requirements. Different researchers have tried to address this problem with some degree of success. The service requests submitted by users at the application layer of cloud framework are realized as tasks in the real environment. One of the major challenges for heterogeneous cloud is how to meet a huge number of heterogeneous tasks while providing the quality of service (QoS) guarantee. Task consolidation is a key technology adopted by heterogeneous clouds to maximize utilization of various resources and use this increased resource utilization to reduce energy consumption. This technique is facilitated by another technology called virtualization which provides the necessary abstraction to the underlying hardware and allow the running of several tasks on a single physical resource concurrently.

This paper considers the problem of finding an energy efficient optimal solution for the allocation of resources in heterogeneous cloud environment. The rest of the paper is organized as follows. Section 2 discusses the related work, Section 3 discusses the cloud framework, Section 4 discusses the problem definition, in Section 5, a generalized system model (including host model, task model and virtual machine model) is introduced and also describes the energy consumption model. To model the heterogeneous computing (HC) environment, ETC (Expected Time to Compute) model is discussed. ETC model [1] expresses the heterogeneity among the run-time of the tasks to be executed and among the machines in the HC environment. Section 6 discusses energy model, then, an Energy Aware Task Consolidation algorithm based on ETC matrix and utilization of various resources (CPU, main memory and disk) are proposed in section 7. Finally the simulation results and analysis show the effectiveness of the algorithm in section 8 followed by conclusion in section 9.

## II. Related Works

Energy consumption is an important issue in heterogeneous cloud and has received more attention because of green computing in trend. Research results show that CPU utilization greatly affects the energy consumption. Many methods have been developed to enhance the utilization of resources in the cloud that include DVFS (Dynamic Voltage Frequency Scaling), memory compression, request discrimination, defining a usage threshold value for resources, task scheduling among virtual machines. One of the key techniques for energy efficient resource allocation is task consolidation [19]. This sec-

tion describes various task consolidation algorithms developed by researchers. In [18], the authors have presented two energy conscious task consolidation algorithms (ECTC and MaxUtil) which aims to maximize utilization of resources and considers both idle and active energy consumption into account where they considered homogeneous computing resources. The algorithms, tries to assign tasks on to the resources for which energy consumption is minimized without any degradation in performance. Energy model is built based on the utilization of resources, CPU is the only resource considered saying that energy consumption is directly proportional to resource utilization. Task processing times are considered as hard deadlines and as the turn OFF/ON of a machine takes a non-negligible amount of time so idle resources are not considered. The results showed that regardless of migration policy, ECTC and MaxUtil outperformed random algorithm by 18% and 13% respectively.

The author in [2] have designed an Enhanced First-t Decreasing Algorithm integrated with VM reuse strategy, DVFS technology and live migration to reduce energy consumption within a data center without violating an SLA in terms of task execution deadline. The algorithm tries to control the best frequency depending on the CPU load. As the load increases, the frequency increase and so is the energy consumption. Thus, depending on the task deadline, frequency is controlled and energy consumption is reduced. For every virtual machine falling below the minimum utilization, the virtual machine with least load that can handle this virtual machine is searched. All the running tasks are migrated onto that machine and the other virtual machine is shut down. Cloud Report was used to simulate the real cloud environment and the performance was compared with greedy and a round robin algorithm. The results showed that proposed EWRR (Enhanced Weighted Round Robin) makes better utilization of resources by consolidating tasks onto a few nodes. The authors in [6] treat communications demands of jobs equally important to that of computational demands and has presented a scheduler called e-STAB.

The authors in [10] proposed a two-state energy-conservation approach, i.e. Power Nap, which simply the complex power performance states of systems. With the help of a power provisioning approach, RAILS (Redundant Array for Inexpensive Load Sharing), they improved power consumption by 74The work in [7] focuses on a batch mode algorithm with the objective of minimizing energy consumption in Heterogeneous Computing Systems (HCS). The system model consists of variably capable machines incorporated with an effective energy saving mechanism for idle time slots. The tasks are considered to be an independent and indivisible work load and the computational model is taken as an ETC model [1]. Simulation is carried out on a set of randomly built ETCs and the algorithm is compared with an existing algorithm minmin. Performance parameters considered for comparison are make-span, flow-time and energy consumption and the results showed that the algorithm behave comparative to min-min but with lower complexity. The work [3] presents an optimization model for task scheduling to minimize task processing time and energy consumption in data center for cloud computing where greedy task scheduling algorithm is proposed for homogeneous tasks. The proposed algorithm is simulated in Matlab and performance is measured based on average task waiting time and total energy consumed by data center versus total number of active servers. The author in [8] has presented an Energy conscious task consolidation technique (ETC) to minimize energy consumption by restricting CPU usage below a specified peak threshold. Energy consumption is separated into two states: idle and running. The task consolidation strategy uses the best-t technique to optimize resources and has defined a 70% CPU utilization as the upper threshold for allocating any virtual machine. Simulation results showed a significant power saving of ETC over recently developed greedy algorithm MaxUtil by 17%. The degree to which the machine execution time varies for a given task is referred as machine heterogeneity and the degree to which task execution times vary for a given machine is referred as task heterogeneity. From the work studied above, the observation is that, most of the research works assumed to be homogeneous system. But in real application systems greatly vary in terms of their resource capabilities. Also the service requests submitted by users vary greatly in terms of their computational and communicational complexities. To the best of our knowledge only a very few researchers have modelled both task heterogeneity and machine heterogeneity in their research. In a work [1] the author has described an ETC (Expected Time to Compute) model to introduce heterogeneity in distributed Heterogeneous computing systems. Based on this, four categories of ETC matrix were proposed in [9].

## III. CLOUD COMPUTING ARCHITECTURE

In this section, we have demonstrated the cloud computing architecture with the help of fig. 1 [12].The Cloud computing model consists of a fully interconnected set of resources. These resources can be physical machines, database servers, network devices, etc. The physical machine or host represents a physical computing node in the cloud with pre-configured resources like CPU, memory, storage, network latency, etc. In this work, the system model constitutes of heterogeneous physical machines that vary greatly in their computing capabilities. As shown in the figure-1, the top layer represents the consumers. The consumers can be either service brokers or the users that submit their service requests at the application layer. The requests submitted are treated as tasks in the cloud during scheduling. So a task is defined as an independent service request made by the user with certain resource requirements and other QoS parameters depending upon the type of service desired. In this model, we have considered that tasks are arriving dynamically into the system. The tasks, then wait in a global queue before they allocate resources. After all tasks are arrived, the next work is performed by Service Scheduler. It is also a physical node and it assigns service requests to virtual machines and determines resource entitlements for allocating virtual machines. The decision of adding or removing virtual

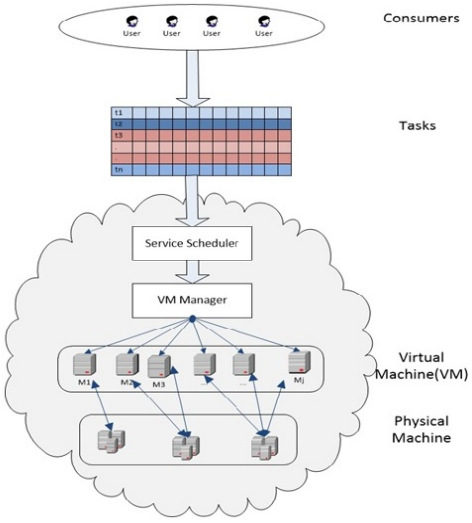


Fig. 1. Cloud Framework

machines according to demand is also taken by the scheduler. The scheduler can be both centralized or distributed depending upon the size of the cloud. Here, a centralized service scheduler is taken for scheduling of tasks. Finally, if a task is meeting all its requirements, it is assigned else it is rejected. The virtual machine is the basic unit to execute a task. Virtual Machine Manager (VMM) will monitor virtual machines as well as the resources.

## IV. PROBLEM STATEMENT

The problem of maximizing resource utilization where minimizing energy consumption is an NP-Complete problem. The problem is a multi-objective problem and the objectives are (1) Minimize energy consumption, (2) Maximize resource utilization, (3) Makespan minimization, (4) Load balancing, (5) Guarantee QoS, (6) Enhance throughput and (7) SLA completion. Here, in this work, minimization of energy consumption while maximizing resource utilization is taken as primary objective. The designed heuristic also tries to minimize the make-span. The system model gives an idea about the nature of physical hosts, their resource capabilities, the interconnections among them. So, the first contribution in this paper defines a proper system model which includes host model and virtual machine model. It is a generalized model and can be used in different scenarios depending upon the application requirements. The host contains all the physical resources required for task implementation in including storage resources, computational resources, network resources and some other hardware devices. The model is described as below:

### A. Host Model

The set H = $\{h_1, h_2, ..., h_m\}$ is defined as the set of physical hosts such that $\mid H \mid$ = m. In this set, each $h_i, i \in [1, m]$ indicates host i, such that $h_i = \{hId_i, hTRes_i, hFRes_i, hTask\_set_i, h\_Vm_i\}$. Here, $hId_i$ is the host identification of host i, $hTRes_i$ is the total resource capability of a host i, $hTRes_i = \{hTR_{i1}, hTR_{i2}, ..., hTR_{ik}\}$ such that $hTR_{ij}, i \in [1, m], j \in [1, k]$ is the total resource capability of $j^{th}$ resource running on $i^{th}$ host. $hFRes_i = \{hFR_{i1}, hFR_{i2}, ..., hFR_{ik}\}$ such that $hFRes_{ij}, i \in [1, m], j \in [1, k]$ is the free resource capability of $j^{th}$ resource running on $i^{th}$ host. $hTask\_set_i$ describes the set of tasks that are allocated to $i^{th}$ host and $h\_Vm_i$ is set of virtual machines that are running on $i^{th}$ host.

### B. Virtual Machine Model

For each host $h_i$, $Vm_i$ is the set of finite virtual machine $Vm_i = \{v_{1i}, v_{2i}, ..., v_{li}\}$ such that $\mid Vi \mid$ = l. Each $v_{ji}, j \in [1, l]$ and $i \in [1, m]$ indicates virtual machine $v_j$ running on host $h_i$. Each $v_{ji}$ is represented by $\{vId, vTRes, vFRes, vFree, vPower\}$. $vId_i$ is the $i^{th}$ virtual machine identification, $vTRes_i$ is the total resource capacity of the $i^{th}$ virtual machine and $vTRes_i = \{vTR_{i1}, vTR_{i2}, ..., vTR_{ik}\}$ such that $vTR_{ij}, i \in [1, l]$ and $j \in [1, k]$ is the total resource capacity of$j^{th}$ resource running on$i^{th}$ virtual machine. $vFRes_i = \{vFR_{i1}, vFR_{i2}, ..., vFR_{ik}\}$ such that $vFRes_{ij}, i \in [1, i]$ and $j \in [1, k]$ is the free resource capability of $j^{th}$ resource running on $i^{th}$ virtual machine. $vFree_i$ is a Boolean variable signifying whether the $i^{th}$ virtual machine is free or not and $vPower_i$ describes the total power consumed by the $i^{th}$ virtual machine. In general, tasks are not homogeneous and varies greatly in their computational requirements. Hence, there is a need of defining a task model that can be easily mapped onto system model.

### C. Task Model

Let's consider T = $\{t_1, t_2, ..., t_n\}$ a set of n independent tasks with random arrival. Each task $t_i$, submitted by the user can have six parameters, i.e. $t_i = \{tId_i, tArr_i, tRes_i, tETC_i, tVm_i, tAss_i\}$. Here, $tId_i$ is the task identification of task $t_i$, $tArr_i$ is the arrival time of task $t_i$, $tRes_i$ is the required resource of the task $t_i$ and it is defined by $tRes_i = \{tR_{i1}, tR_{i2}, ..., tR_{ik}\}$ such that $tR_{ij}$, $i \in [1, n]$ and$j \in [1, k]$ is the requirement of resource$R_j$ by the task $t_i, tETC_i$ defines the ETC (Expected Time to Compute) matrix for task $t_i$, which is a $1m$ matrix (m being the number of hosts), $tVm_i$ is the type of virtual machine required by task $t_i$, $tAss_i$ is a boolean variable representing whether task is scheduled or not.

## V. LPP FORMULATION OF TASK CONSOLIDATION

According to the above definitions, a linear programming problem formulation for the Task Consolidation problem is given. The LPP formulation of the problem is given below:

- **Minimize** $E(0,t) = \sum_{j=1}^{m} \sum_{i=1}^{l} e_{ij}(0,t)$
- Subject to :
  1) $\sum_{i=1}^{l} e_{ij} \leq e_j \quad \forall j \in [i, m].$
  2) $\sum_{i=1}^{l} vTRes_j \leq hTRes_j \quad \forall j \in [i, m].$
  3) $\sum_{i=1}^{l} vFRes_j \leq hFRes_j \quad \forall j \in [i, m].$

where E(0,t) describes the total energy consumed by the cloud in the time interval [0,t]. $e_{ij}(0,t)$ represents the energy consumed by virtual machine i running on host j in time interval [0,t]. The first constraint restricts the total energy consumed by all the virtual machine inside a host to be less than the energy consumed by that host. Second constraint says that the total resources of all the virtual machines running on a host should always be less than the total resources of that host. Similarly, third constraint states that total free resources of all the virtual machines running on a host should always be less than the total available resources of that host. All these conditions must hold true at every instant of time.

### A. Solution to Task Consolidation Problem

To design an energy efficient solution for task consolidation, certain assumptions are to be defined for this defined model. Tests include both the computational time as well as communications time. It means that ETC value will be the total time taken by any task on a machine. Also, all the tasks are independent and heterogeneous. This specific assumption model heterogeneity among tasks because in real time scenario tasks vary greatly in their computational complexities and other resource requirements. All the tasks are considered as non-preemptive in nature. Arrival time is considered to be Poison distribution. All the systems are heterogeneous in terms of their resource capabilities. It models system heterogeneity because in actual system varies greatly in terms of their processor speed, RAM size and other resource capabilities. A task is allowed to execute only on a single machine. Also the other overheads like start and shutdown time of virtual machines are considered to be constant and the last is that all virtual machines are installed on all physical hosts.

Table : 1 Heterogeneity table

| Case | Low | High |
|---|---|---|
| $T\_Hetro$ | 10 | $10^5$ |
| $M\_Hetro$ | 10 | $10^2$ |

### B. Modeling Heterogeneity of Cloud computing environment

To better evaluate the mapping of tasks onto machines in a heterogeneous computing environment, a model is required that performs well even when computing environment changes. One such model consists of an ETC (Expected time to Compute) matrix which has been discussed in [1] and the same is used in this research work. According to this model, four categories of ETC matrix were proposed in [9] which are described below:

1) High Task Heterogeneity and High Machine Heterogeneity
2) High Task Heterogeneity and Low Machine Heterogeneity
3) Low Task Heterogeneity and High Machine Heterogeneity
4) Low Task Heterogeneity and Low Machine Heterogeneity

In ETC matrix, entry (i, j) indicates the execution time of the task i on machine j. A range based method and a coefficient-of-variation based method to generate ETC matrix are discussed, the later one providing a greater control over the spread of values. Range based method is used for simulation purpose.

## VI. ENERGY MODEL

The energy model discussed here is derived from [8]. According to it, the energy consumption of any virtual machine is separated in two states: Idle state and Running state. To compute energy they have only considered CPU, but this work also dealing with other resources(RAM and disk). So, the energy is computing based on the mean utilization of all the resources. The 6 different levels are described as below:

- $\propto W$ if idle
- $\beta$ + $\propto W$ if $0\% < CPU$ utilization$\leq 20\%$
- $3\beta$ + $\propto W$ if $20\% < CPU$ utilization$\leq 50\%$
- $5\beta$ + $\propto W$ if $50\% < CPU$ utilization$\leq 70\%$
- $8\beta$ + $\propto W$ if $70\% < CPU$ utilization$\leq 80\%$
- $11\beta$ + $\propto W$ if $80\% < CPU$ utilization$\leq 90\%$
- $12\beta$ + $\propto W$ if $90\% < CPU$ utilization$\leq 100\%$

## VII. TASK CONSOLIDATION ALGORITHMS

Task consolidation is an effective means to efficiently utilize resources in a cloud leading to significant improvement in energy consumption of data center. So, based on the above energy model, an Energy Aware Task consolidation (EATC) algorithm is designed that tries to allocate the incoming task onto the machine that takes minimum time for executing that task. Energy is computed based on the mean utilization of a virtual machine.

**Algorithm 1:** *ETC_Generation*

**Input**: $n\_host, T\_hetro, M\_hetro$
**Output**: An ETC matrix of order $[1 n\_host]$
**begin**
  **Compute** $a = \cup(1, T\_hetro)$
  **for** *i 0 to* $(n\_host - 1)$ **do**
    $b = \cup(1, M\_hetro)$
    $ETC[1, i] = ab$
  **end**
  Return $ETC$
**end**

**Algorithm 2:** *Resource_Generation*

**Input**: $n\_vms$
**Output**: Task along with its resource requirements i.e. CPU,RAM,Disk,$vm\_type$
**begin**
  **Compute** CPU = $\cup(x, y)$
  **Compute** RAM = $\cup(x, y)$
  **Compute** Disk = $\cup(x, y)$
  **Compute** **vm_type** $= \cup(1, n\_vms)$
  Return All_resources : CPU,RAM,Disk,**vm_type**
**end**

The description of algorithm is as follows: The algorithm takes to generate the task arrival at every time unit using

Poisson distribution. The for every incoming task, two functions named $Resource\_Generation$ and $ETC\_Generation$ are called that generates the required resources (CPU, RAM, disk, virtual machine type) and ETC matrix respectively. ETC matrix is sorted and algorithm tries to assign every task on the host for which ETC value is minimum. After every allocation, virtual machine utilization is updated and energy consumed is calculated. After every allocation, hTask set is updated. If no resource is able to fulfil a task requirement, the task is rejected. Finally, total energy consumed and hTask set is returned.

**Algorithm 3:** EATC

**Input**: $n\_host, T\_hetro, M\_hetro, st, \lambda, n\_vms$
**Output**: $hTask\_set, Energy$
**begin**
  $n\_tasks$ = 0.
  **for** $i = 1$ *to* $st$ **do**
    $a = Poission(\lambda)$
    **for** $j = 1$ *to* $a$ **do**
      $n\_tasks = n\_tasks + 1$
      $Required\_resources =$
      $Resource\_Generation(n\_vms)$
      $ETC[n\_tasks] =$
      $ETC\_Generation(n\_host, T\_hetro, M\_hetro)$
      $Sorted\_host = Sort(ETC[n\_tasks])$
    **end**
  **end**
  **for** $k = 1$ *to* $n\_tasks$ **do**
    **for** $l = 1$ *to* $n\_host$ **do**
      **for** $m = 1$ *to* $n\_host$ **do**
        **if** $Sorted\_host(1, l) == ETC(1, m)$
        **then**
          **if** *the required virtual machine of* $m^{th}$
          *host can full the task requirements*
          **then**
            $h_l Task\_set = h_l Task\_set \cup tId_k$
            Update the available resources
            Update Energy Consumed
            Break.
          **end**
        **end**
      **end**
      **if** *task assigned* **then**
        Break.
      **end**
    **end**
    **if** *task not assigned* **then**
      Reject the task.
    **end**
  **end**
  Return $Energy, hTask\_Set$
**end**

## VIII. SIMULATION RESULTS

The simulation is carried out with the simulator designed using Matlab [3], [13]. Simulation time was taken to be 20 seconds. The simulation was carried out for 3 types of arrival rates, namely low traffic arrival, moderate traffic arrival, high traffic arrival. For each traffic type, number of tasks arrived were computed using a Poisson distribution.

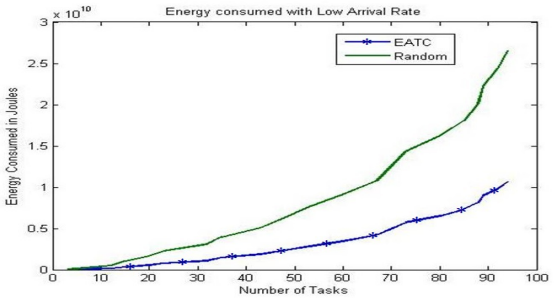


Fig. 2. Energy consumption at low arrival rate of number of tasks with Low Task Heterogeneity and Low Machine Heterogeneity

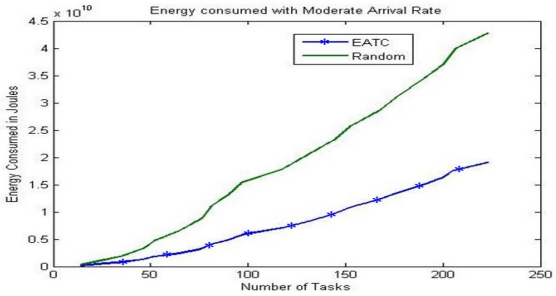


Fig. 3. Energy consumption at moderate arrival rate of number of tasks with Low Task Heterogeneity and Low Machine Heterogeneity

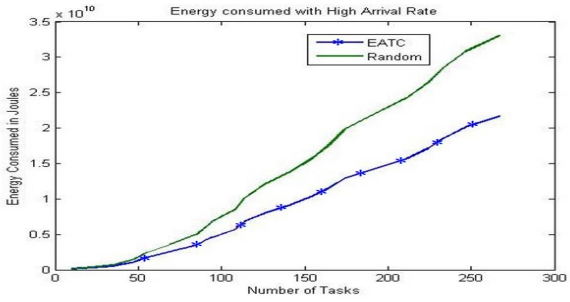


Fig. 4. Energy consumption at high arrival rate of number of tasks with Low Task Heterogeneity and Low Machine Heterogeneity

$\lambda$ value was passed as 5, 10, 15 for low, moderate, high traffic rate respectively. There are 50 number of hosts and for each host, 3 virtual machines are taken and their resources capabilities are generated using a uniform distribution between 1 and 1000. For each hosts, 3 resources, including CPU, ram, disk are assumed. Task requirements are also generated using uniform distribution in range 1 to 100. The value for $\alpha$ and $\beta$ are taken as 5 Joules and 10 Joules respectively. For all the four types of ETC matrix, the values of $T\_Hetro$ and $M\_Hetro$ are taken according to table 1. Finally, energy was

computed based on the utilization of resources. Unit of energy measurement is taken to be in Joules. The graph is plotted between the number of tasks arrived and the total energy consumed in the cloud. For each traffic arrival rate, 4 graphs are generated i.e, one for each ETC matrix. Therefore, a total of 12 graphs are obtained out of which 6 graphs from Fig. 2 to Fig. 7 are shown here and the graph shows that this model consumed less energy.

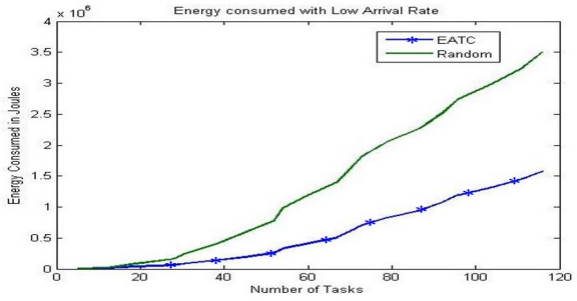


Fig. 5. Energy consumption at low arrival rate of number of tasks with High Task Heterogeneity and Low Machine Heterogeneity

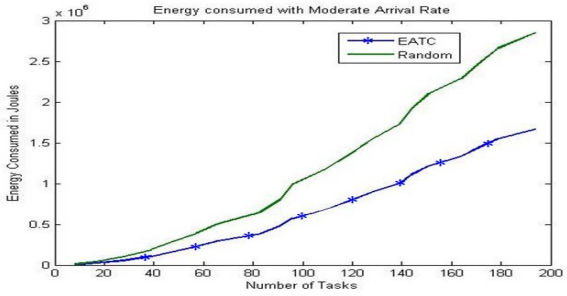


Fig. 6. Energy consumption at moderate arrival rate of number of tasks with High Task Heterogeneity and Low Machine Heterogeneity

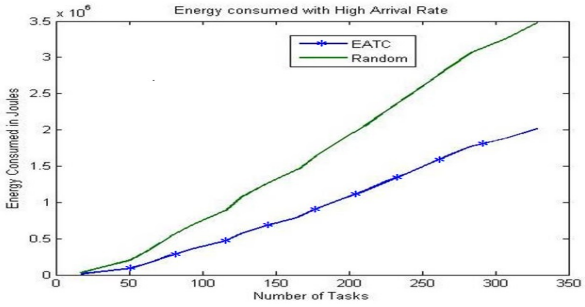


Fig. 7. Energy consumption at high arrival rate of number of tasks with High Task Heterogeneity and Low Machine Heterogeneity

## IX. Conclusion

This paper studies the task consolidation problem in a heterogeneous cloud computing environment. A system model, including host model, virtual machine model, and task model is proposed that takes into account ETC model proposed by [1]. For the proposed models, EATC (Energy Aware Task Consolidation) algorithm is developed that tries to allocate the tasks on the machines for which energy consumption is minimized. The performance is analyzed against a random scheduler for different arrival rate of tasks (low, moderate, high). Evaluation is done for the number of tasks versus total energy consumed by those tasks and the results showed a significant improvement in energy savings.

## References


[1] Ali, Shady, et al. "Task execution time modeling for heterogeneous computing systems." Heterogeneous Computing Workshop, 2000.(HCW 2000) Proceedings. 9th. IEEE, 2000.

[2] Alnowiser, Abdulaziz, et al. "Enhanced weighted round robin (ewrr) with dvfs technology in cloud energy-aware." Computational Science and Computational Intelligence (CSCI), 2014 International Conference on. Vol. 1. IEEE, 2014.

[3] Liu, Ning, Ziqian Dong, and Roberto Rojas-Cessa. "Task and server assignment for reduction of energy consumption in datacenters." Network Computing and Applications (NCA), 11th IEEE International Symposium on. IEEE, 2012.

[4] Luo, Liang, et al. "A resource scheduling algorithm of cloud computing based on energy efficient optimization methods." Green Computing Conference (IGCC), 2012 International. IEEE, 2012.

[5] T Ograph, B., and Y. Richard Morgens. "Cloud computing." Communications of the ACM 51.7 (2008).

[6] Kliazovich, Dzmitry, et al. "e-STAB: energy-efficient scheduling for cloud computing applications with traffic load balancing." Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing. IEEE, 2013.

[7] Diaz, Cesar O., et al. "Energy-aware fast scheduling heuristics in heterogeneous computing systems." High Performance Computing and Simulation (HPCS), 2011 International Conference on. IEEE, 2011.

[8] Hsu, Ching-Hsien, et al. "Optimizing energy consumption with task consolidation in clouds." Information Sciences 258, 452-462, 2014.

[9] Armstrong Jr, Robert K. Investigation of effect of different run-time distributions on SmartNet performance. Naval postgraduate school monterey ca, 1997.

[10] Meisner, David, Brian T. Gold, and Thomas F. Wenisch. "PowerNap: eliminating server idle power." ACM Sigplan Notices. Vol. 44. No. 3. ACM, 2009.

[11] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing.", 2011.

[12] Kumar, Dilip, and Bibhudatta Sahoo. "Energy Efficient Heuristic Resource Allocation for Cloud Computing." 2014.

[13] Pranitha, P., and A. Rathinam. "Load Balancing of Grid Connected Data Centers Using Various Optimization Techniques." Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on. IEEE, 2012.

[14] Chen, Huangke, et al. "ERES: An Energy-Aware Real-Time Elastic Scheduling Algorithm in Clouds." High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on. IEEE, 2013.

[15] Jiang, Jing. "Optimised auto-scaling for cloud-based web service.", 2015.

[16] Srikantaiah, Shekhar, Aman Kansal, and Feng Zhao. "Energy aware consolidation for cloud computing." Proceedings of the 2008 conference on Power aware computing and systems. Vol. 10. 2008.

[17] Valentini, Giorgio L., Samee U. Khan, and Pascal Bouvry. "Energy-efficient resource utilization in cloud computing." Large Scale Network-centric Computing Systems, John Wiley & Sons, Hoboken, NJ, USA. 2013.

[18] Lee, Young Choon, and Albert Y. Zomaya. "Energy efficient utilization of resources in cloud computing systems." The Journal of Supercomputing 60.2, 268-280. 2012.

[19] Panda, Sanjaya K., and Prasanta K. Jana. "An Efficient Resource Allocation Algorithm for IaaS Cloud." Distributed Computing and Internet Technology. Springer International Publishing, 351-355. 2015.