# An Efficient Energy Consumption Minimizing Based on Genetic and Power Aware Scheduling in Cloud Computing

**M Asma Parveen[1], Jilcy[2], Sanimol S[3], Senthil Prabhu S[4]**

*[1,2,3]Student, Dept of Computer Science and Engineering, Dhaanish Ahmed Institute of Technology, Tamil Nadu, India*
*[4]Professor, Dept of Computer Science and Engineering, Dhaanish Ahmed Institute of Technology, Tamil Nadu, India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract** - *One of the keys faced challenges in this field of cloud computing is how to reduce the energy consumption and power consumption in cloud computing data centers. Which affects the resource utilization and economic benefits. This paper solves the problem such as reducing power and energy consumption with minimum cost which improves the utilization of resources where it process under workload independent quality of service constraints. The Dynamic Single Threshold algorithm(DST) VM consolidation leverages fine-grained fluctuations within the application workloads and unceasingly reallocates VM mistreatment migration to attenuate the amount of active physical nodes. A genetic algorithm based power aware scheduling of resources allocates(G-PARS) has been proposed to solve the dynamic virtual machine allocation policy problem. Results of the experiment using DST and G-PARS shows that the proposed algorithm minimize the power consumption compared with existing (MECABP) algorithm under different scale conditions.*

**KeyWords:** Dynamic Single Threshold, Energy consumption, G_PARS, VM allocation .

## 1. INTRODUCTION

This document is template. We ask that authors follow some simple guidelines. In essence, we have a tendency to raise you to create your paper look precisely like this document. The easiest thanks to do that is just to transfer the model, and replace(copy-paste) the content with your own material. Number the reference things consecutively in sq. brackets (e.g. [1]). However the authors name is used along side the reference range within the running text. The order of reference within the running text ought to match with the list of references at the tip of the paper.

### 1.1 Cloud services overview

Cloud service models describe however services square measure created out there to users. We distinguish between 2 differing types of models: classic Cloud service models and new hybrid ones.

Infrastructure as a Service (IaaS)
Platform as a Service (PaaS)
Software as a Service (SaaS)

### 1.2 Classic Cloud service models

Classic Cloud service models will be classified into 3 types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and package as a Service (SaaS).

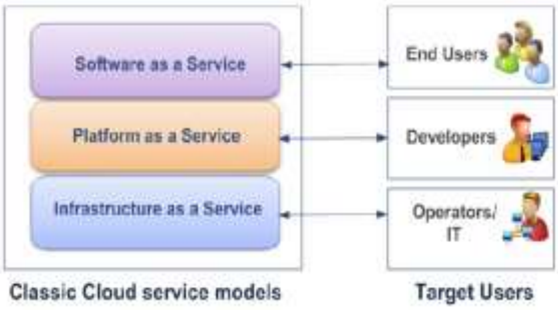


**Figure 2.2**Classic Cloud service models

### 1.3 Energy consumption challenge in clouds

The numerous advantages of cloud computing environments, including cost effectiveness, on-demand scalability, and ease of management, encourage service providers to adopt them and offer solutions via cloud models. This in turn encourages platform providers to increase the underlying capacity of their data centers to accommodate the increasing demand of new customers. One of the most drawbacks of the expansion in capability of cloud information centers is that the would like for a lot of energy to power these large-scale infrastructures. Such a drastic growth in energy consumption of cloud data centers is a major concern of cloud providers.

### 1.4 Resource optimization algorithms

#### 1.4.1 Genetic Algorithm (GA)

This is based on evolutionary model for complex systems. This algorithm is a global, efficient and parallel searching technique. It is like a natural evolution method. Therefore, GA is well suitable for solving the resource optimization problem. Multiple point searching is an enhanced feature of genetic algorithm. The search space is reduced by using GA techniques in resource optimization. In this approach, the candidate's solutions are represented in terms of chromosomes which are mended in each stage. GA uses the selection methodology to find the survival of fitness function. For each and every stage the highest fitness value is found and those values are inherited to next generation.

### 1.4.2 Ant Colony Optimization (ACO) Algorithm

It is a randomized probabilistic technique, and used to find an optimal route which is similar to the ants to find the route for their food. Generally, ants produce a pheromone when they find a route. This route is attracted by more ants that directs to find the optimal route. In cloud computing environment a cloud service provider provides different cloud resources on different virtual machines. Here ACO plays a vital role for optimization of cloud resources. Virtual machines are considered as nodes, ants are assumed as agents and resources are similar to the food. Ant travels among the nodes and assign the requested jobs to the cloud resources.

### 1.4.3 Particle Swarm Optimization (PSO) Algorithm

It is a computational method which belongs to the super set of swarm's intelligences. In this algorithm is self adaptive9 iterative procedure is used to optimize a problem. The candidate solutions ar} iteratively processed with relation to a given measure of quality. In PSO, population of candidate solutions is said as 'particles. The optimisation in PSO involves movement of those particles around within the search-space. The movement is target-hunting by easy mathematical formulae over the particle's rate and position. Each particle's local best-known position influences its movement. In addition to the present it's additionally target-hunting toward the known positions within the search-space, which are updated as better positions are found by other particles. The swarm is meant to converge towards the best resolution by these particle movements

### 1.4.4 Bacterial Foraging Optimization (BFO) Algorithm

Bacterial foraging algorithm is based on hyper-heuristics8 method. In this algorithm, partial result is denoted by bacterium and the motion of the bacterium as heuristics. The optimization in bacterial foraging algorithm lies on process like chemo taxis, swarming, reproduction, elimination and dispersal. The aim of the algorithm is how the animal finds the food is similar to that the power consumption per time (P/T) is increased. Chemo taxis: This method is based on the motion of the Escherichia Coli bacteria through drowning and breaking down via flagella. If the bacterium is going in the similar way for a time period. The movement of the bacteria towards the foods but it can either be pulled or declined. The bacterium will attract the different kinds of bacteria throughout the times.

### 1.5 Power-aware resource management

There is a large body of literature investigating energy management techniques for PaaS cloud service model that provides a platform for cloud customers to develop, run, and manage their applications without worrying about the underlying infrastructure and the required software. Both kinds of virtualization namely, OS level and System level virtualization, are considered and the newly introduced CaaS model can be viewed as a form of OS level virtualization service. Since CaaS cloud model has been newly introduced, we grouped all the research with the focus on containerized (OS-level virtualized) cloud environments under the PaaS category.

Servers are one of the most power-hungry elements in data centers, with CPU and memory as their main power consumers. The average power consumption of CPU and memory is reported to be 33% and 23% of the server's total power consumption respectively. Therefore, any improvement on processor and memory-level power consumption would definitely reduce the total power consumption of the server, which also improves the energy efficiency of data center. Dynamic voltage and frequency scaling (DVFS) are an effective system level technique utilized both for memory and CPU in bare metal environments and it is demonstrated to improve the power consumption of these two elements considerably. DVFS enables dynamic power management through varying the supply voltage or the operating frequencies of the processor and/or memory.

## 2. Literature Review

[1] T. Kumrai, et., al., "Multi objective optimization in cloud brokering systems for connected Internet of Things," 2017In this paper, Currently, over nine billion things are connected in the Internet of Things (IoT). This variety is anticipated to exceed20 billion within the close to future, and the number of things is quickly increasing, indicating that numerous data will be generated. It is necessary to make AN infrastructure to manage the connected things. [2] M. A. Rodriguez and R. Buyya, "Deadline based totally resource provisioning and programming rule for scientific workflows on clouds," 2014 Cloud computing is that the latest distributed computing paradigm and it offers tremendous opportunities to solve large scale scientific problems. However, it presents varied challenges that require to be self-addressed so as to be expeditiously utilised for work flow applications. [3] H. Arabnejad and J. G. Barbosa, "A budget constrained scheduling algorithm for workflow applications," 2014Service-oriented computing has enabled a new method of service provisioning based on utility computing models, in which users consume services based on their Quality of Service (QoS) requirements. In such pay-per-use models, users square measure charged for services supported their usage and on the fulfilment of QoS constraints; execution time and value square measure 2 common QoS needs. [4] W.

Chen, et., al., "Efficient task programing for budget strained parallel applications on heterogeneous cloud computing systems," 2017 because the cost-driven public cloud services emerge, budget constraint is one in all the primary style problems in large-scale scientific applications dead on heterogeneous cloud computing systems. Minimizing the schedule length whereas satisfying the budget constraint of AN application is one among the foremost necessary quality of service needs for cloud suppliers. [5] H. Arabnejad, et., al., "Low-time complexity budget deadline constrained workflow scheduling on heterogeneous resources," 2016The execution of scientific applications, under the utility computing model, is constrained to Quality of Service (QoS) parameters. Commonly, applications have time and price constraints such all tasks of an application have to be compelled to be finished at intervals a user-specified point and Budget.

## 3. Design methodology

### 3.1 Problem definition

- Resource allocation is performed considering computational and networking requirements of tasks and optimizes task completion time and data center power consumption.
- The propose algorithm is design using genetic algorithms that allow both to explore solutions space and to search for the optimal solution in an efficient manner.

### 3.2 PROPOSE METHODOLOGY

The dynamic Merging of Virtual Machines (VMs) in Cloud data centers. The goal is to enhance the employment of computing resources and scale back energy consumption beneath work freelance quality of service constraints. Dynamic VM consolidation leverages fine-grained fluctuations within the application workloads and unendingly reallocates VMs victimization live migration to attenuate the amount of active physical nodes. Energy consumption is reduced by dynamically deactivating and reactivating physical nodes to fulfill the present resource demand. Scientific workflows is composed of the many fine procedure roughness tasks. The runtime of those tasks is also shorter than the length of system overheads, as an example, once mistreatment multiple resources of a cloud infrastructure. Minimizing the energy consumption in cloud computing surroundings is one amongst the key analysis problems.'

•The propose dynamic single Threshold (DST), is based on the idea of dynamic an upper utilization threshold for hosts and placing VMs while keeping the total utilization of the CPU below this threshold.
•Minimization of Migrations (MM)
•Highest Potential Growth (HPG) – migrating VMs that have rock bottom usage of central processor comparatively to the requested so as to reduce total potential increase of the use and SLA violation. Policy under an explicitly specified QoS goal for any known stationary workload and a given state configuration in the online setting.
•Random Choice (RC)
•Energy-Aware Task Scheduler using Genetic Algorithm.

This algorithm can meet the minimum resource requirement of a business and refrain the extra consumption of energy hence, we can decrease the energy utilization in data centers.

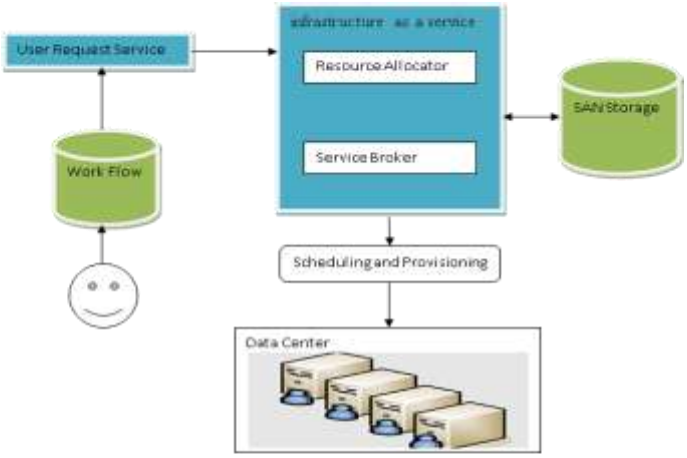


**Figure 3.2**Architecture diagram

### Algorithm: Dynamic Utilization Thresholds

```
Input: hostList, vmList Output: migrationList
vmList.sortDecreasingUtilization()
foreach h in hostList do
hUtil ←h.util()
bestFitUtil← MAX
while hUtil>h.upThresh() do
foreach vm in vmList do
if vm.util() > hUtil − h.upThresh() then
t ← vm.util()− hUtil + h.upThresh()
if t < bestFitUtil then
bestFitUtil← t
bestFitVm← vm
else
if bestFitUtil = MAX then
bestFitVm← vm
break
hUtil← hUtil − bestFitVm.util()
migrationList.add(bestFitVm)
vmList.remove(vm)
if hUtil<lowThresh() then
migrationList.add(h.getVmList())
vmList.remove(h.getVmList())
return migrationList
```

The advantage of grouping the information for every VM individually and so victimisation the summation is that a VM is migrated along side the information of its resource usage and the data are actual even once a VM migration. Using this information and the inverse cumulative probability function for the t-distribution (t_(inv_n ) (P))it is possible to find out an interval of the CPU utilization, which will be reached with a low probability (e.g. 5%). It can set the upper utilization threshold (T_ui ) for each host i preserving this amount of spare CPU capacity defined by the lower (P_ul ) and upper (T_uu ) limits of the chance interval, where n is that the variety of information points collected, and n-1represents the degrees of freedom for the t-distribution.

$$T_{ui} = 1 - \left(\left(t_{inv_{n-1}}(P_{uu}) \cdot SU_i + \bar{U}_i\right) - \left(t_{inv_{n-1}}(P_{ul}) \cdot SU_i + \bar{U}_i\right)\right) \qquad (6)$$

The lower threshold is calculated in a similar way; how-ever, the difference is that a single value is obtained for all the hosts in the system. The idea is to see the hosts that have lower utilizations comparatively to the average value across all the nodes. To tackle the case when all the hosts have low CPU utilizations, we introduce a limit (U_l ) to cap the decrease of the lower utilization threshold. To calculate the lower threshold (T_l )

$$\bar{U} = \frac{1}{N}\sum_{i=1}^{N} \bar{U}_i\,, \qquad S_U = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(\bar{U}_i - \bar{U})^2}$$

$$T_l = \begin{cases} \bar{U} - t_{inv_{n-1}}(P_l) \cdot S_U & if < U_l \\ U_l & otherwise \end{cases} \qquad (7)$$

The DT algorithm apply the MM policy for VM selection, as in our previous work it has shown the superiority over the alternatives. The complexness of the algorithmic program is proportional to the total of variety the amount the quantity of non-over-utilized host and the merchandise of the amount of over-utilized hosts and also the number of VMs allotted to those over-utilized hosts.

The VM placement is seen as a bin packing drawback with variable bin sizes and costs, where bins represent the physical nodes; items are the VMs that have to be allocated; bin sizes are the obtainable processor capacities of the nodes; and costs correspond to the facility consumption by the nodes. As the bin packing problem is NP-hard, to solve it apply a modification of the Best Fit Decreasing (BFD) algorithm that is shown to use no more than 11=9 · OPT + 1 bins. In our modification (MBFD)

### Algorithm: Modified Best Fit Decreasing (MBFD)

```
Input: hostList, vmList Output: allocation of VMs
vmList.sortDecreasingUtilization()
foreach vm in vmList do
minPower← MAX
allocatedHost← NULL
foreach host in hostList do
if host has enough resource for vm then
power ← estimatePower(host, vm)
if power < minPower then
allocatedHost← host
minPower← power
if allocatedHost≠ NULL then
allocate vm to allocatedHost
return allocation
```

Sort all the VMs in the decreasing order of current CPU utilizations and allocate each VM to a host that provides the least increase of the power consumption caused by the allocation. This allows the leveraging the nodes heterogeneity by choosing the most power-efficient ones first.VM allocation will be done using Genetic approach. But initially VMs will be assigned to random set of physical machines.

## 4. CONCLUSION

One of the keys faced challenges in this field is how to reduce the massive amount of energy consumption in cloud computing data centers. To address this issue, several power-aware virtual machine (VM) allocation and consolidation approaches area unit projected to scale back energy consumption with efficiency. However, most of these existing economical cloud solutions save energy value at a value of the numerous performance degradation. The propose a unique technique for genetic primarily based dynamic consolidation of VMs supported adaptational utilization thresholds, that ensures a high level of meeting the Service Level Agreements (SLA). The evaluated the proposed algorithms through extensive simulations on a large-scale experimental setup using workload traces. The experiments show that our planned strategy incorporates a higher performance than different ways, not only in high QoS but also in less energy consumption. In addition, the advantage of its reduction on the amount of active hosts is far a lot of obvious, particularly once it's below extreme workloads..

## 5. FUTURE ENHANCEMENT

For the future work, we propose to investigate the focusing on multi-core CPU architectures, as well as consideration of multiple system resources, such as memory and network interface, as these resources additionally considerably contribute to the energy consumption. In order to judge the planned system in an exceedingly real Cloud infrastructure, we have a tendency to decide to implement it by extending a real-world Cloud platform. Besides the reduction in infrastructure and on-going operational prices, this work additionally has social significance because it decreases CO2 footprints and energy consumption by trendy IT infrastructures.

## REFERENCES

[1] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, ``Multi-objective optimization in cloud brokering systems for connected Internet ofThings,'' IEEE Internet Things J., vol. 4, no. 2, pp. 404-413, Apr. 2017.

[2] K. Xie et al., ``Distributed multi-dimensional pricing for efficient application offloading in mobile cloud computing,'' IEEE Trans. Services Comput., to be published, doi: 10.1109/TSC.2016.2642182.

[3] M. A. Rodriguez and R. Buyya, ``Deadline based resource provisioningandscheduling algorithm for scientific workflows on clouds,'' IEEE Trans.Cloud Comput., vol. 2, no. 2, pp. 222-235, Apr./Jun. 2014.

[4] G. Xie et al., ``Minimizing redundancy to satisfy reliability requirement for a parallel application on heterogeneous service-oriented systems,'' IEEE Trans. Services Comput., to be published, doi: 10.1109/TSC.2017.2665552.

[5] H. Arabnejad and J. G. Barbosa, ``A budget constrained schedulingalgorithm for workflow applications,'' J. Grid Comput., vol. 12, no. 4,pp. 665-679, 2014.

[6] W. Chen, G. Xie, R. Li, Y. Bai, C. Fan, and K. Li, ``Efficient task scheduling for budget constrained parallel applications on heterogeneous cloudcomputing systems,'' Future Generat. Comput. Syst., vol. 74, pp. 1-11,Sep. 2017.

[7] H. Arabnejad and J. G. Barbosa, ``List scheduling algorithm for heterogeneous systems by an optimistic cost table,'' IEEE Trans. Parallel Distrib.Syst., vol. 25, no. 3, pp. 682-694, Mar. 2014.

[8] H. Arabnejad, J. G. Barbosa, and R. Prodan, ``Low-time complexity budgetdeadline constrained workflow scheduling on heterogeneousresources,'' Future Generat. Comput. Syst., vol. 55, pp. 29-40, Feb. 2016.

[9] G. Xie, Y. Chen, X. Xiao, C. Xu, R. Li, and K. Li, ``Energy-efficientfault-tolerant scheduling of reliable parallel applications on heterogeneousdistributed embedded systems,'' IEEE Trans. Sustain. Comput., to bepublished, doi: 10.1109/TSUSC.2017.2711362.

[10] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, ``Deadline-constrained workflow scheduling algorithms for infrastructure as a serviceclouds,'' Future Generat. Comput. Syst., vol. 29, no. 1, pp. 158-169, 2013.

[11] T. Wei et al., ``Cost-constrained QoS optimization for approximatecomputation real-time tasks in heterogeneous MPSoCs,'' IEEE Trans.Comput.-Aided Des. Integr. Circuits Syst., to be published, doi: 10.1109/TCAD.2017.2772896.

[12] H. Li, M. Dong, K. Ota, and M. Guo, ``Pricing and repurchasing for big data processing in multi-clouds,'' IEEE Trans. Emerg. Topics Comput., vol. 4,no. 2, pp. 266-277, Apr./Jun. 2016.

[13] S. Wang, Z. Qian, J. Yuan, and I. You, ``A DVFS based energy-efficienttasks scheduling in a data center,'' IEEE Access, vol. 5, pp. 13090-13102,2017.

[14] W. Huang, Z. Wang, M. Dong, and Z. Qian, ``A two-tier energy-awareresource management for virtualized cloud computing system,'' Sci. Programm., vol. 2016, p. 6, Oct. 2016.

[15] M. Mao and M. Humphrey, ``Auto-scaling to minimize cost and meet application deadlines in cloud workflows,'' in Proc. Int. Conf. High Perform.Comput., Netw., Storage Anal. (SC), Nov. 2011, pp. 1-12.

[16] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, ``Cost-driven scheduling of grid workflows using partial critical paths,'' IEEE Trans. ParallelDistrib. Syst., vol. 23, no. 8, pp. 1400-1414, Aug. 2012.

[17] C. Q. Wu, X. Lin, D. Yu, W. Xu, and L. Li, ``End-to-end delayminimization for scientific workflows in clouds under budget constraint,'' IEEE Trans. Cloud Comput., vol. 3, no. 2, pp. 169-181,Apr./Jun. 2015.

[18] Y. C. Lee and A. Y. Zomaya, ``Energy conscious scheduling fordistributed computing systems under different operating conditions,''IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 8, pp. 1374-1381,Aug. 2011.

[19] X. Xiao, G. Xie, R. Li, and K. Li, ``Minimizing schedule length of energyconsumption constrained parallel applications on heterogeneous distributed systems,'' in Proc. IEEE Trustcom/BigDataSE/ISPA, Aug. 2016,pp. 1471-1476.

[20] Q. Huang, S. Su, J. Li, P. Xu, K. Shuang, and X. Huang, ``Enhanced energyefficient scheduling for parallel applications in cloud,'' in Proc. 12thIEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGrid), May 2012,pp. 781-786.

[21] G. Xie, J. Jiang, Y. Liu, R. Li, and K. Li, ``Minimizing energy consumptionof real-time parallel applications using downward and upward approacheson heterogeneous systems,'' IEEE Trans. Ind. Informat., vol. 13, no. 3,pp. 1068-1078, Mar. 2017.