Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 11, November 2018, pg.204 – 215

An Heuristics based Dynamic Power-Aware Resource Allocation for Cloud Computing

D.Vignesh[1], A.Finny Belwin[2], A.Linda Sherin[3], Dr. Antony Selvadoss Thanamani[4]

[1]Research Scholar Department of Computer Science & Bharathiar University, India
[2]Research Scholar Department of Computer Science & Bharathiar University, India
[3]Research Scholar Department of Computer Science & Bharathiar University, India
[4]Professor and Head Department of Computer Science NGM College, Pollachi, India
[1] vigneshudt100@gmail.com; [2] belwin35@gmail.com; [3] linz15sherin@gmail.com; [4] selvdoss@gmail.com

Abstract— Cloud computing is public pools of configurable mainframe system resources and higher-level services that can be rapidly provisioned with minimal running effort, often over the Internet. One of the main challenges in cloud computing is how to reduce the massive amount of energy consumption in cloud computing data centers. The many research authors proposed power aware resource allocation algorithm to solve this issue based on virtual machine allocation and consolidation approaches. The most of existing energy efficient cloud solutions save energy cost at a price of the significant performance degradation. In this paper propose a genetic heuristic search optimization technique based dynamic consolidation of VMs based on adaptive utilization thresholds, which ensures a high level of meeting the service level agreements (SLA).The dynamic virtual machine allocation policy heuristics based on the idea of setting upper and lower utilization thresholds for hosts and keeping total utilization of CPU by all VMs between these dynamic changing thresholds. The power-aware scheduling-based resource allocation (G-PARS) has been proposed to solve the dynamic virtual machine allocation policy problem. The experiments result shows that the proposed strategy has a better performance than particle swarm optimization strategies, not only in high QoS but also in less energy consumption. In addition, the advantage of its reduction on the number of active hosts is much clearer, especially when it is under life-threatening workloads.

Keywords— ***Cloud computing, Resource Allocation, Dynamic Utilization Threshold, Power Aware Scheduling.***

I. Introduction

Cloud computing has become one of the fastest growing paradigms in computer science. It is a model for giving IT assets as an administration in a cost-effective and pay-per-utilize way. Distributed computing is a compensation for every utilization display for empowering helpful, on-request arranges access to a common pool of configurable registering assets, for example, systems, servers, stockpiling, applications, and administrations. An important aspect to consider with the Cloud is the ownership and use of the Cloud infrastructure [1]. Diverse methodologies can be utilized to convey Cloud frameworks:

- Private cloud: The private cloud infrastructures owned and managed by a single company, used in a private network and not available for public use.
- Community cloud: The community to shared cloud infrastructures for specific communities composed by multiple users.
- Public cloud: Refers to superior and substantial foundations worked by outside organizations that give IT administrations to numerous shoppers by means of the Internet.
- Hybrid cloud: As the name as of now demonstrates, a hybrid cloud is a mix of both a private and open cloud. Parts of the administration keep running on the organization's private cloud, and parts are re-appropriated to an outer public cloud.

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 [20]. One of the fundamental downsides of the development in limit of cloud server farms is the requirement for more vitality to control these huge scale frameworks. Such an extraordinary development in vitality utilization of cloud server farms is a noteworthy worry of cloud suppliers.

Energy wastage in data centers are driven by various reasons such as inefficiency in data center cooling systems, network equipment's, and server utilization. However, servers are still the main power consumers in a data center [3]. Both the amount of work and the efficiency with which the work is performed affect the power consumption of servers. Therefore, for improving the power efficiency of data centers, the energy consumption of servers should be made more proportional to the workload.

Power proportionality is defined as the proportion of the amount of power consumed comparing to the actual workload and it can be achieved by either decreasing servers' idle power utilization at hardware level or efficient provisioning of servers through power-aware resource management policies at software level. Although there is a large body of research on energy efficient resource management of IaaS, not enough attention has been given to PaaS environments with containers. Hence, this thesis focuses on software-level energy management techniques that are applicable to containerized cloud environments [2]. The main objective is improving data center energy consumption while maintaining the required Quality of Service (QoS) through decreasing SLA violations. This thesis contributes to the literature by considering both containerized and enterprise cloud environments while addressing their new challenges. One of the perspectives that recognizes this proposition from the related work is that this theory handles the issue of server farm vitality utilization through the investigation of genuine enterprise cloud backend data[19]. It also explores the potential benefits, for enterprise and containerized cloud environments, from a comprehensive cloud workload study and how it can decrease the amount of energy consumption in the data center.

The contribution of this work to reduce power consumption of Data centers is an important issue because of large amount of electricity consumption:

1. The resource allocation of VMs is carried out in two steps: at the first step select VMs that need to be migrated, at the second step chosen VMs are placed on hosts using modified best fit decreasing;
2. The heuristic, Single Threshold (ST), is based on the idea of setting upper utilization threshold for hosts and placing VMs while keeping the total utilization of CPU below this threshold; and
3. Genetic algorithm for power-aware in scheduling of resource allocation (G-PARS) has been proposed to solve the dynamic virtual machine allocation problem.

Rest of the paper is organized as follows, section I contain overview of cloud computing and different type of cloud deployment. Section II contain review of exiting cloud scheduling and power ware resource algorithms, Section III contain proposes system and module implementations, Section IV contain result and discussion , performance analysis , Section V concludes.

II. RELATED WORK

The existing PSO algorithm has been used to place VMs; it is still a great challenge to consolidate VMs to PMs across multiple data centers for saving power. The DPRA mechanism based on PSO is first proposed to allocate more kinds of resources than and to consolidate VMs across multiple data centers, which attempts to minimize the energy consumption of PMs and air conditioners and the electric bill of data centers. The DPRA yields a lower total energy consumption compared with the PSO [4]. A versatile asset control framework that progressively changes the asset offers to singular levels with the end goal to meet application-level QoS objectives while accomplishing high asset use in the Datacenter. Our control framework is produced utilizing established control hypothesis, and we utilized a discovery framework displaying way to deal with conquer the nonappearance of first rule models for complex endeavor applications and systems.

Proficient and reliable work process planning (WFS) is urgent for coordinating endeavor frame works. While WFS has been widely studied, WFS-related algorithms are mainly focused on optimizing execution time or cost [18]. Be that as it may, in cloud computing condition, WFS is up against the dangers of the inborn vulnerability and untrustworthy to the applications. Hence, trust benefit arranged systems must be considered in WFS. Integer Linear Programming (ILP) is the issue plan premise. The Tree algorithm is proposed to put VM job occurrences at the most reduced correspondence cost, conserving the development cost with less physical servers. Another Forest algorithm is likewise proposed for adjusting the calculation stack between the physical machines.

A power-efficient VN provisioning problem as a mathematical optimization problem, with the objective of minimizing the power consumption by employing mixed-integer programming develop a set of heuristics that prevent overload in the system effectively while saving energy used. Trace driven simulation and experiment results demonstrate that our algorithm achieves good performance [17].The design, implementation, and evaluation of a resource management system for cloud computing services. Auction-based models for VM provisioning and allocation which allow users to submit bids for their requested VMs[7]. We define the dynamic VM provisioning and assignment issue for the bartering based model as a whole number program thinking about various kinds of resources. We at that point plan honest eager and ideal components for the issue with the end goal that the cloud provider provisions VMs dependent on the solicitations of the triumphant clients and decides their installments.

The existing eco-Cloud, a self-organizing and adaptive approach for the consolidation of VMs on two resources, namely CPU and RAM. Choices on the task and movement of VMs are driven by probabilistic procedures and depend solely on neighborhood data, which makes the methodology exceptionally easy to actualize. The existing author to work is two-fold: first, we present multiple ways to capture the cost-aware application placement problem that may be applied to various settings. For every definition, we give points of interest on the sort of data required to take care of the issues, the model presumptions, and the common sense of the suspicions on genuine servers. In the second part of our study, we present the mapped architecture and placement algorithms to solve one practical formulation of the problem: minimizing power subject to a fixed performance requirement [6]. The VM migration problem in cloud data center is formulated based on mixed integer linear programming, and the VM Allocation algorithm is proposed to construct a stable, robust, balanced network.

Virtual Resource Management Protocol (VRMP) that specifies set of mechanisms to be followed and describe messages to be exchanged while creating and deploying virtual clusters. It has been implemented as services namely Virtual Cluster Service (VCS), Virtual Machine Service (VMS) and Virtual Resource Aggregation Service (VRAS) for virtual resource management and monitoring across grid environment. The investigated power-aware provisioning of VMs for soft and hard real-time Cloud services. For hard real-time services, we have provided several schemes and evaluated those using simulations [8]. For delicate constant administrations, we have dissected power-aware productive VM provisioning and proposed a provisioning algorithm. Incoming workload requests initiated by application users' (or load generator) gets queued into workload dispatcher (or load balancer module) in first come first served (FCFS) basis. Workload dispatcher is an intelligent arbitrator, which controls and schedules the request to a physical server Si which would consume the least power for the application.

In a data center environment, there is a clear trade-off between leaving idle servers on, and thus minimizing mean response time with no significant power saving, versus turning idle servers off or putting them to SLEEP state, which hurts response time but may save power. Incoming workload requests initiated by application users' (or load generator) gets queued into workload dispatcher (or load balancer module) in first come first served (FCFS) basis. Workload dispatcher is an intelligent arbitrator, which controls and schedules the request to a physical server Si which would consume the least power for the application. For simplicity, we assume all the physical servers are of homogeneous configuration and have one SLEEP state in addition to other possible states {BUSY, IDLE, OFF}. Physical server in BUSY state indicates that the server has in process requests running in one or more of its VMs' [9]. Physical server in IDLE state indicates that the server has no in process requests running in its VMs'. Physical server in SLEEP state indicate that the server is in reduced power state (HIBERNATE or SUSPEND state). Physical server in OFF state indicates that the server is in switched off state. Physical server in SETUP state indicates that the server is transitioning between states for example, SLEEP to IDLE state or OFF to IDLE state.

The problem of reducing energy consumption is an important concern from operational expense perspective as the cloud environment (embodiment of virtualized environment) grows in size and complexity. Existing work deal with physical SLEEP state transition impacts to overall power consumption and response times in a physical server environment. In this work, we have proposed a heuristic approach in accounting for SLEEP state at server level and using VMs' to processing application requests virtualized environment. Results from this exercise shows savings to overall average per application request response time and marginal workload specific power consumption savings on certain scenarios.

III. PROPOSED METHODOLOGY

The propose power consumption by data center can be accurately described by a linear relationship between the power consumption and CPU utilization, even when dynamic voltage and frequency scaling (DVFS) is applied. The reason lies in the set number of states that can be set to the recurrence and voltage of CPU and the way that voltage and execution scaling are not connected to other framework parts, such as memory and network interface. Moreover, these implementations show that on average an idle server consumes approximately 70% of the power consumed when it is fully utilized.

The power utilization as a component of the CPU usage (P (u)) as shown in (1).

$$p(u) = k \cdot P_{max} + (1 - k) \cdot P_{max} \cdot u = P_{max} \cdot (0.7 + 0.3.u) \qquad (1)$$

Where P_{max} is set to 250 W, which is a usual value for modern computing servers; k

is the fraction of power consumed by an idle server; and u is the CPU utilization. As the utilization of CPU may change over time due to the workload variability, it is a function of the time: u (t). Therefore, to define the total energy consumption by a server we use the model defined in (2).

$$E = \int_t P(t(t)) \ dt \qquad (2)$$

As per this model, the vitality utilization by a server is dictated by the CPU usage. Hence, to decrease the vitality utilization, our methodology is to enhance the CPU usage of physical hubs in a data center.

A. Cloud Model

The CloudSim is executed at the following dimension by automatically broadening the center functionalities uncovered. CloudSim gives novel help to demonstrating and reproduction of virtualized Cloud based server farm conditions, for example, devoted administration interfaces for VMs, memory, stockpiling, and transfer speed. CloudSim layer deals with the instantiation and execution of center elements (VMs, has, server farms, application) amid the recreation time frame. This layer is able to do simultaneously instantiating and straightforwardly dealing with a substantial scale Cloud foundation comprising of thousands of framework segments. The major issues, for example, provisioning of hosts to VMs dependent on client demands, overseeing application execution, and dynamic checking are taken care of by this layer. A Cloud supplier, who needs to the viability of various power portion arrangements in allotting its hosts, would need to execute his techniques at this layer by automatically broadening the center VM provisioning usefulness. There is an unmistakable qualification at this layer on how a host is distributed to various contending VMs in the Cloud. A Cloud host can be simultaneously shared among various VMs that execute applications dependent on client characterized QoS details.

B. Data Centre Resource Allocation

Dynamic Single Threshold (DST), depends on setting an upper use edge for hosts and putting VMs while keeping the aggregate usage of the CPU underneath this limit. Enhancement of current assignment of VMs is completed in two stages: at the initial step we select VMs that should be relocated, at the second step picked VMs are set on hosts utilizing hereditary calculation. We propose four heuristics for picking VMs to relocate. The principal heuristic, Single Threshold (ST), depends on setting upper use limit for hosts and putting VMs while keeping the aggregate use of CPU beneath this threshold [10]. The point is to protect free assets to avert SLA infringement because of solidification in situations when use by VMs increments. At each time allotment all VMs are reallocated utilizing hereditary calculation with extra state of keeping the upper usage limit not disregarded. The new situation is accomplished by live movement of VMs.

The other three heuristics depend on setting upper and lower usage limits for hosts and keeping complete use of CPU by all VMs between these edges. In the event that the usage of CPU for a host goes beneath the lower edge, all VMs must be moved from this host and the host must be turned off with the end goal to kill the inactive power utilization. In the event that the usage goes over the upper limit, some VMs must be relocated from the host to lessen use with the end goal to counteract potential SLA infringement. We propose three approaches for picking VMs that must be relocated from the host.

- Minimization of Migrations (MM) – relocating minimal number of VMs to limit movement overhead.

- Highest Potential Growth (HPG) – relocating VMs that have the most minimal use of CPU generally to the asked for with the end goal to limit add up to potential increment of the use and SLA infringement.

- Random Choice (RC) – picking the essential number of VMs by picking them as per a consistently conveyed arbitrary variable.

C. VM selection and placement

VM assignment can be separated in two: the initial segment is the confirmation of new demands for VM provisioning and putting the VMs on hosts, while the second part is the enhancement of the current VM distribution. The initial segment can be viewed as a container pressing issue with variable receptacle sizes and prices[12][13]. The Modified Best Fit Decreasing (MBFD) calculations, we sort all VMs in diminishing request of their present CPU usages, and allot each VM to a host that gives minimal increment of intensity utilization because of this portion. This permits utilizing the heterogeneity of assets by picking the most power-effective hubs first. The pseudo-code for the calculation is displayed in Algorithm. The intricacy of the designation part of the calculation is $n \cdot m$, where n is the quantity of VMs that must be distributed and m is the quantity of hosts.

The streamlining of the current VM assignment is completed in two stages: at the initial step we select VMs that should be moved, at the second step the picked VMs are set on the hosts utilizing the MBFD calculation. To decide when and which VMs ought to be moved, we present three twofold edge VM choice strategies. The essential thought is to set upper and lower usage limits for hosts and keep the aggregate use of the CPU by all the VMs designated to the host between these edges. On the off chance that the CPU usage of a host falls beneath the lower limit, all VMs must be relocated from this host and the host must be changed to the rest mode with the end goal to dispense with the inactive power utilization. On the off chance that the usage surpasses the upper limit, some VMs must be relocated from the host to decrease the use. The point is to save free assets with the end goal to avoid SLA infringement because of the solidification in situations when the use by VMs increments.

D. Algorithm Implementation

First of all, it assumes that the CPU utilization created by each VM can be described by a random variable (with a particular distribution, which persists at least over some recent period of time. In this case, the CPU utilization of a host can be represented by a random variable (, which is a sum of utilizations by m VMs allocated to this host. Here assume that as the distribution created by different VMs are different, the distribution of the host's utilization is approximately normal and can be modelled by the t-distribution[14][15]. It cannot predict the CPU utilization of a physical node in the future; however, we can calculate characteristics of the distribution over some recent period of time, such as the sample mean (and standard deviation (S

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 collecting the data for each VM separately and then using the summation is that a VM is migrated together with the data of its resource usage and the information will be genuine even after a VM migration. Using this information and the inverse cumulative probability function for the t-distribution $(t_{inv_n}($ 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 $(T$ for each host i preserving this amount of spare CPU capacity defined by the lower $(P$ and upper $(T_i$ limit of the probability interval, where n is the quantity of information focuses gathered, and n-1 speaks to the degrees of opportunity for the t-dispersion.

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

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 thought is to decide the hosts that have bring down usages generally to the normal incentive over every one of the hubs. To tackle the case when all the hosts have low CPU utilizations, we introduce a limit $($ to cap the decrease of the lower utilization threshold. To calculate the lower threshold $($

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

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

The DT algorithm apply the MM policy for VM selection, as in our previous work it has shown the superiority over the alternatives. The multifaceted nature of the algorithm is relative to the total of the quantity of non-over-used host in addition to the result of the quantity of over-used hosts and the quantity of VMs allocated to these over-utilized hosts.

The VM arrangement can be viewed as a container pressing the issue with bin sizes and costs, where receptacles speak to the physical hubs; things are the VMs that need to be allocated; bin sizes are the available CPU capacities of the nodes; and prices correspond to the power consumption by the nodes[16]. 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.

Algorithm of GA

i. randomly initializes population(t)

ii. determine fitness of population(t)

iii. repeat

a. selects parents from population(t)

b. perform crossover on parents creating population(t+1)

c. performs mutation of population(t+1)

d. determines fitness of population(t+1)

iv. until best individual is good enough

Then according to the performance and situation fitness of solution will be calculated and then this solution if having fitness more than threshold will take part in crossover and mutation.

IV. RESULTS AND DISCUSSION

The data centre that comprises 10 heterogeneous physical nodes. Each node is modelled to have one CPU core with performance equivalent to 2000, 2500, 3000 or 3500 Million Instructions Per Second (MIPS), 16 GB of RAM, 10 GB/s network bandwidth and 1 TB of storage. Power consumption by the hosts is defined by the model. As indicated by this model, a host devours from 175 W with 0% CPU usage up to 250 W with 100% CPU use. Each VM requires one CPU centre with most extreme of 1000, 2000, 2500 or 3250 MIPS, 1 GB of RAM, 100 Mb/s arrange data transmission and 1 GB of capacity. Notwithstanding, amid the lifetime VMs, may utilize fewer resources making the open door for a dynamic combination. The CPU MIPS ratings are equivalent to cloudsim instance types. The clients submit demands for provisioning of 500 heterogeneous VMs. Each VM has haphazardly relegated an outstanding task at hand follow from one of the servers from the remaining burden information. At first, VMs are apportioned by their parameters expecting 100% use.

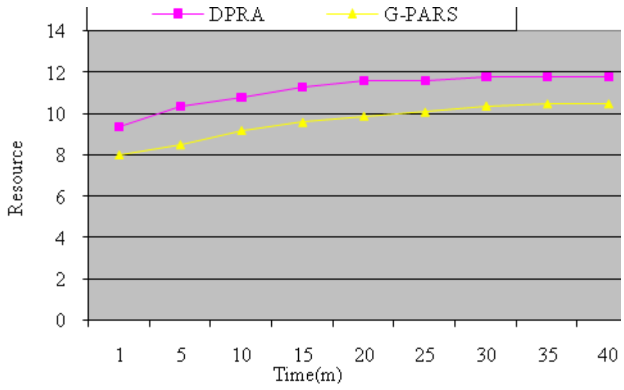

Figure 1. Comparison of resource utilization existing with proposed system

The figure 1 show the time vs resource allocation in existing system take high time for resource allocation, the proposed consume less time.

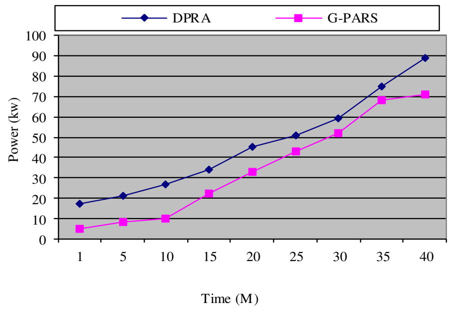

Figure2. Comparison of powers utilization existing with proposed system

The figure 2 show the better power reduction of each data centre compare with existing algorithm. The propose power aware resource allocation take less power consumption.

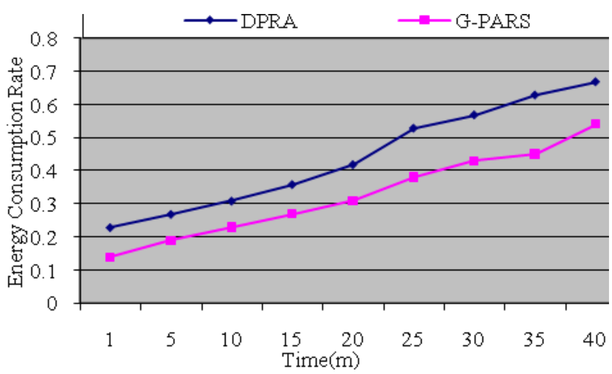

Figure 3. Comparison of energy utilization existing with proposed system

Figure 3 show the energy consumption in this propose system less energy consumption compare with existing algorithm.

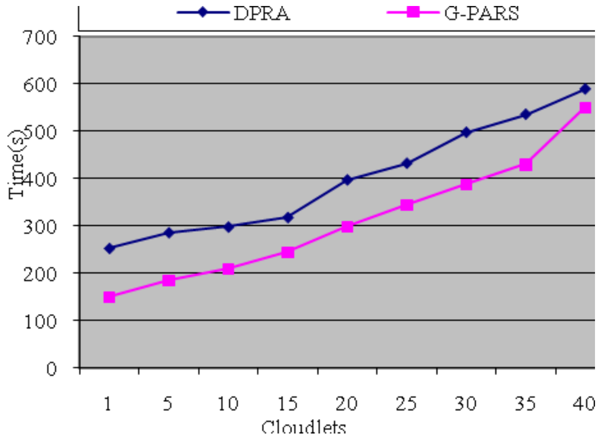

Figure 4. Comparison of cloud job finishing time existing with proposed system

Figure 4 shows the jobs vs. finishing time the propose system take less time to finish the all jobs.

V. CONCLUSION

In this paper to solve how to reduce the massive amount of energy consumption in cloud computing data centre. To address this issue, many power-aware virtual machine (VM) allocation and consolidation approaches are proposed in exiting system to reduce energy consumption efficiently. However, most of those existing efficient cloud solutions save energy cost but not consider the price level. The propose a search optimization technique based dynamic consolidation of VMs based on adaptive utilization thresholds, which ensures a high level of meeting the service level agreements (SLA). The evaluated the proposed algorithm through extensive simulations on a large-scale experimental setup using workload traces. The tests demonstrate that our proposed system has a superior execution than different methodologies, in high QoS as well as in less vitality utilization.

REFERENCES

[1] P. Padala et al., "*Adaptive control of virtualized resource in utility computing environments,"* in Proc. ACM SIGOPS/EuroSys Conf. Comput. Syst., Mar. 2007, pp. 289–302.

[2] P. C. Hershey, S. Rao, C. B. Silio, and A. Narayan, *"System of systems for quality-of-service observation and response in cloud computing environments,"*IEEE Syst. J., vol. 9, no. 1, pp. 212–222, Mar. 2015.

[3] W. A. Tan et al., "*A trust service-oriented scheduling model for workflow applications in cloud computing,"* IEEE Syst. J., vol. 8, no. 3, pp. 868–878, Sep. 2014.

[4] F.-H. Tseng, C.-Y. Chen, L.-D. Chou, H.-C. Chao, and J.-W. Niu, *"Serviceoriented virtual machine placement optimization for green data center,"* ACM/Springer Mobile Netw. Appl., vol. 20, no. 5, pp. 556–566, Oct. 2015.

[5] C.-C. Lin, H.-H. Chin, and D.-J. Deng, *"Dynamic multiservice load balancing in cloud-based multimedia system,"* IEEE Syst. J., vol. 8, no. 1, pp. 225–234, Mar. 2014.

[6] R. Yamini, *"Power management in cloud computing using green algorithm,"* in Proc. Int. Conf. Adv. Eng., Sci. Manage., Mar. 2012, pp. 128–133.

[7] G. Sun, H. Yu, V. Anand, D. Liao, and L. Li, *"Exploring power-efficient provisioning for online virtual network requests,"* in Proc. Int. Conf. Comput. Inf. Technol., Oct. 2012, pp. 51–55.

[8] G. Sun et al., *"Power-efficient provisioning for online virtual network requests in cloud-based data centers,"* IEEE Syst. J., vol. 9, no. 2, pp. 427– 441, Jun. 2015.

[9] Y.-W. Ma, J.-L. Chen, C.-H. Chou, and S.-K. Lu, "*A power saving mechanism for multimedia streaming services in cloud computing,"* IEEE Syst. J., vol. 8, no. 1, pp. 219–224, Mar. 2014.

[10] S. E. Dashtia and A. M. Rahmani, *"Dynamic VMs placement for energy efficiency by PSO in cloud computing,"* J. Exp. Theor. Artif. Intell., vol. 28, pp. 1–16, Apr. 2015.

[11] Z. Xiao, W. Song, and Q. Chen, *"Dynamic resource allocation using virtual machines for cloud computing environment,"* IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1107–1117, Jun. 2013.

[12] M. M. Nejad, L. Mashayekhy, and D. Grosu, *"Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds,"* IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 2, pp. 594–603, Feb. 2015.

[13] C. Mastroianni, M. Meo, and G. Papuzzo, *"Probabilistic consolidation of virtual machines in self-organizing cloud data centers,"* IEEE Trans. Cloud Comput., vol. 1, no. 2, pp. 215–228, Jul.–Dec. 2013.

[14] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, *"Green cloud computing: balancing energy in processing, storage, and transport,"* Proc. IEEE, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[15] R. Buyya, A. Beloglazov, and J. Abawajy, *"Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges,"* in Proc. Int. Conf. Parallel Distrib. Process. Tech. Appl., Jul. 2010, pp. 1–12.

[16] A. Verma, P. Ahuja, and A. *Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems,"* in Proc. ACM/IFIP/USENIX Int. Conf. Middleware, Dec. 2008, pp. 243–264.

[17] Iosup, A., Ostermann, S., Yigitbasi, M.N., Prodan, R., Fahringer, T. and Epema, D.H.J, Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing, IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, 2011, pp. 931-945.

[18] Junaid Qayyum, Faheem Khan, Muhammad LaL, Fayyaz Gul, Muhammad Sohaib and Fahad Masood, Implementing and Managing framework for PaaS in Cloud Computing, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No. 3, 2011, pp. 474-479.

[19] Thilagavathi, D. and A. S. Thanamani, A *Survey on Dynamic Job Scheduling in Grid Environment Based on Heuristic Algorithms, International Journal of Computer Trends and Technology* Vol. 3, Issue 4, 2012, pp. 531-536.

[20] Thilagavathi, D. and A. S. Thanamani, Heuristics in Grid Scheduling, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* Vol. 2 Issue 8, August 2013, pp. 2427-2432.

[21] Johny Antony P, Dr. Antony Selvadoss Thanamani, "A Privacy Preservation Framework for Big Data (Using Differential Privacy and Overlapped Slicing)", International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 10, October 2016

[22] R.NANDHAKUMAR1 AND ANTONY SELVADOSS THANAMANI, "A Survey on E-Health Care for Diabetes Using Cloud Framework", International Journal of Advanced Research Trends in Engineering and Technology (IJARTET) Vol. 4, Issue 10, October 2017

[23] N.Raveendran1 Dr. Antony Selvadoss Thanamani, "Autonomic Cloud Services to Enhance Secure Data Sevices for Digital Library", American Research Journal of Computer Science and Information Technology (ARJCSIT) Volume 1, 10 pages

[24] Ms. D. Thilagavathi and Dr. Antony Selvadoss Thanamani August 2014, Scheduling in High Performance Computing Environment using Firefly Algorithm and Intelligent Water Drop Algorithm, International Journal of Engineering Trends and Technology (IJETT) – Volume 14 Number 1

