2013 IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing

# Energy-efficient Scheduling Algorithms for Data Center Resources in Cloud Computing

Tamal Adhikary, Amit Kumar Das, Md. Abdur Razzaque
Green Networking Research Group
Department of Computer Science & Engineering
University of Dhaka, Dhaka - 1000, Bangladesh
Email: tamal.csedu@gmail.com, amit.csedu@gmail.com
razzaque@cse.univdhaka.edu

A. M. Jehad Sarkar
Dept. of Digital Information Engineering
College of Engineering
Hankuk University of Foreign Studies
Yongin-si, Gyeonggi-do, South Korea
Email: jehad@hufs.ac.kr

***Abstract*—A significant amount of energy is consumed to render high-level computation tasks in large scale cloud computing applications. The state-of-the-art energy saving techniques based on centralized job placement approaches reduce the reliability of operation due to a single point of failure. Moreover, the existing works do not consider energy consumption cost for communication devices and network appliances which contribute a lot. In this paper, we have proposed a mechanism for cluster formation based on network vicinity among the data servers. We have developed two distributed and localized intra-cluster and inter-cluster VM scheduling algorithms based on energy calculation, resource requirement and availability. Our proposed scheduling algorithms manage VMs to reduce the energy consumption of both the servers and networking devices. Simulation results show that our proposed distributed VM scheduling algorithms can conserve significant amount of energy compared to state-of-the-art works.**

## I. Introduction

Cloud computing is one of the major forces changing the IT landscape, which is a new model for the dynamic provisioning of computing services supported by cloud data centers that renders VM (Virtual Machine) to the customers. It employes pay-as-you-go model for delivering infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) [1]. A number of big computing service providers such as Google, Microsoft, Yahoo, Amazon and IBM are continuously deploying data centers in various locations around the world.

Nowadays, application scheduling is widely used as an effective energy conservation method. The energy consumption can be significantly reduced by consolidating applications on as less servers as possible and making idle servers sleep or power-off [3]. By optimizing the power consumption of the network infrastructure, few of existing application scheduling has been considered. The network infrastructure is built to provide high bisection bandwidth for applications in typical commodity large-scale systems. By sharing diverse applications and each utilizing a part of the system, every time many of these networks remain unused.

Most of the state-of-the-art works [2], [4] did not consider runtime scheduling of VMs. Since the workload in cloud computing environment varies with time, realtime VM scheduling could reduce energy expenses for computation when workload declines and more VMs should be assigned when the workload increases. Hierarchical arrangement of different physical servers in a data center can increase the bandwidth usage in some cases when two clusters communicate through an upper level connector of the hierarchy [2]. Some other works, e.g. [4], did not consider bandwidth usage during the distribution of VMs within the physical servers. As bandwidth cost incurs significant overhead to the big data centers, considering communication energy and bandwidth consumption among VMs during VM scheduling could reduce the energy consumption and bandwidth requirement to a great extent. None of them exploits local resource availability in scheduling and thus gives poor system performance and increases bandwidth cost to a great extent.

Most of the works on energy optimization in cloud data centers did not provide a complete architecture and arrangement of physical servers in a data center. A few works on infrastructure modeling which made groups of physical servers, did not provide a clear idea about the arrangement and working model of the components inside the groups and the working components which operate and control the servers of each group.

In this paper, we focus on analyzing application scheduling through VM and its impact on the energy usage of the cloud infrastructure in a data center. We have modeled the cloud infrastructure containing clusters of servers. The model of each cluster organization and work division of the components inside the clusters is also designed. To schedule VMs within a cluster and among the clusters within the whole data center, we have developed two distributed VM scheduling algorithms. Our proposed intra- and inter-cluster scheduling algorithms reduce energy consumption by turning off redundant servers and keeping under-utilized clusters in sleep mode. Preference in migrating communicating VMs to same cluster ensures bandwidth utilization and reduces dealy for communication. The efficiency and the stability of our algorithms are verified by simulating it inside a data center using CloudSim toolkit [15].

The rest of the paper is organized as follows. The Section II describes works related to our topics of interest. In section III, netwrok architecture and an energy-efficient cluster organization model have been described. In Section IV, the proposed distributed Intra- and Inter-Cluster VM Scheduling Algorithms

978-0-7695-5088-6/13 $31.00 © 2013 IEEE
DOI 10.1109/HPCC.and.EUC.2013.244

have been presented. The Section V presents the result of performance evaluation and in Section VI, we conclude the paper along with the future research directions.

## II. Related Works

The deployment of cloud computing is emerging day by day. To reduce the maintenance and management cost in comparison with in-house infrastructure, it is becoming attractive and its use is growing rapidly. For supporting the elasticity and scalability required by the customers, cloud providers also rely on large and power-consuming data centers. Energy-efficiency still remains a serious problem for providers, though one of its commercial credentials is the reduction of energy consumption at the client edge [5]. They have dealt with energy consumption as well as provided SLA (Service Level Agreement) and high performance expectations. Energy-efficiency is still a challenging issue in the cloud model due to the dynamicity of its components and flexibility of services supported by virtualization where in the same time it has been widely explored for similar architectures such as Cluster and Grid Computing.

For resizing the pool of servers dynamically, energy-efficient cloud computing approaches of first generation have been introduced [6]. The researchers of this paper have proposed to resize the number of active servers according to the actual demand, by exploiting technologies such as Wake-on-Lan (WoL), Dynamic Voltage Frequency Scaling (DVFS), and live migration. In paper [7], the authors have proposed methodology to enhance this "Dynamic Resource Resizing" (DRR) approach, for maintaining the expected QoS level. By introducing mechanisms to mitigate the overhead caused by VM provisioning, turning on/off servers and executing the large-scale live migrations, the authors aim to reduce the overall SLA violation. Though these approaches represent a very important step to improve the energy efficiency and performance relationship, they are focused on host and data center level improvements neglecting fine-grained concerns such as those at VM level.

In heterogeneous computing environment, some approaches have introduced for scheduling tasks with load-balancing technique. In [8], the authors have introduced some heuristics to schedule tasks for heterogeneous computing nodes. To address which tasks in a task queue have to be run in internal cloud and which tasks can be sent to external cloud, the researchers of [9] proposed a scheduling algorithm. The main focus of the approach is to keep the order of tasks in the queue while increasing performance by utilizing an external cloud on demand. How many and which class of cloud providers are required, how much data is allocated to each chosen cloud for parallel processing are not considered in their paper.

For efficiently over allocating the available host resource, the over allocation algorithm is proposed in [10]. Over allocation model is mainly inherited from overbooking theory in [11], which guides cloud providers confirming the booking of more resources than the amount they actually have to support the service. However, the main benefit for over allocation algorithm is only for the SaaS layer, not for the IaaS layer.

## III. Network Model and Assumptions

The cloud service providers often fail to cope with the increasing costs for rendering services and to mitigate the increasing expenses; in some cases, they have to violate the SLA. Though many models for managing servers and deploying VMs is prevalent now-a-days, energy efficient server management model is scarce and many of them are not suitable for practical deployment. Again most of the models do not take the bandwidth cost into account. Network devices contribute largely to the energy expense of data centers. Communication aware VM scheduling can reduce the operational energy expenditures and bandwidth requirements. We predict resource utilization for the incoming requests using LPF [12]. Transferring data between two nodes connected directly to the same switch tends to use less energy than that of the nodes connected indirectly [13]. Our proposed algorithm prioritizes data transfers among those nodes which are on the same switch, in support of energy efficiency and delay minimization.

### A. Network Model

We have divided the cloud service providers into smaller units called clusters, as shown in Figure 1. Each cluster has its own data storage, computational and resource management unit that allocates VMs to the requests. Each cluster is connected to its adjacent neighbors that can share their computational usage whenever necessary. The VM migration or job distribution becomes faster in a clustered environment. Whenever jobs can't be distributed within a cluster, help from adjacent neighbors can be taken based on their usage. Service downtime for migration of VMs within a cluster or among the clusters is negligible since migration is done on-the-fly. That is, copy of the VM is created first and once the complete copy of the currently working VM is made in the new host server, user is given the new VM.

Each cluster and each server within the cluster will be in one of the two states: $sleep$ and $active$. The properties of the states are:

- *Active mode:* In active mode, a cluster is fully functional. All its servers and the resource manager are working and serving the user requests.
- *Sleep mode:* In sleep mode, a whole cluster can shut down its entire computational, database and other

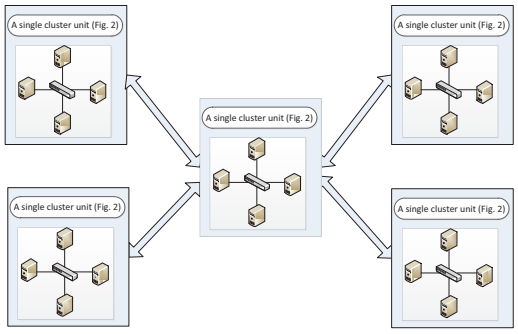



Fig. 1. Multicluster data center environment

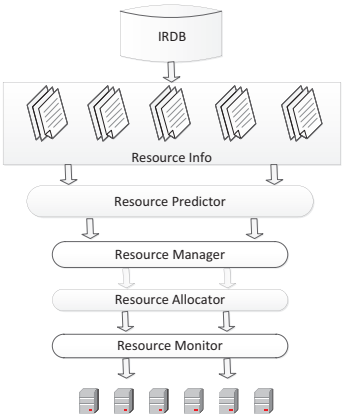



Fig. 2. A single cluster architecture

component servers. Only the communication maintenance server needs to be working at that period.

### B. Cluster Architecture

In Figure 2, a single cluster architecture is shown. The working steps and the activities of each component in a cluster for serving a new request have been given in Algorithm 1.

In each cluster, there is an Information Record Data Base (IRDB) that stores the record of resource requirement of previous tasks. Whenever new request appears at the cluster, IRDB checks for the resource requirement of the request. It then sends the resource information found for the job to the Resource Predictor (RP). The task of resource predictor is to estimate the type and quantity of resources required to serve the given request. Then the resource manager checks for the resource usage at the cluster and determines whether it should provide resource to the request from the cluster or take computational help from one of its neighbors. If resource requirement of the given request can be supplied by the resource capacity of the cluster, then it requests the resource allocator to assign resources to the give request. Resource capacity specifies the amount of resources in the cluster that can be supplied to the users for their request whereas the resource usage of the cluster is within the predefined limit. Resource usage reveals the amount of resources used in a cluster. Finally, the resource allocator assigns Virtual Machines (VM) to the requests. The resource allocator maintains a dynamic pool of VMs for provisioning to new requests. The mechanism of maintaining a dynamic pool has been described in our previous work [14]. The resource requirement for the requests is monitored continuously by the component, resource monitor and whenever the requirement changes, the whole process of VM allocation is recycled. Whenever two VMs in two different cluster transfer data continuously and the communication link between the clusters get congested, two VMs are located in a single cluster, which can serve the two VMs and whose resource utilization is less.

## IV. Proposed Scheduling Algorithms

Whenever the cumulative resource requirement of a cluster goes below a threshold level, it tries to migrate all of its

**Algorithm 1** *VM Provisioning Algorithm*

**INPUT**: *Previous LPF value, IRDB info*
**OUTPUT**: *Resource optimized VM provisioning*

```
1.  while new request appears do
2.    Check IRDB for previous resource utilization info
3.    Calculate LPF using equation 1 and IRDB info
4.    Check for available resources
5.    if resources available then
6.      if VM of predicted resource capacity is available
        then
7.        Allocate VM to the requesting client
8.      else
9.        Create a new VM and serve the request
10.     end if
11.   else
12.     Transfer request to another cluster
13.   end if
14. end while
```

serving requests to a neighboring cluster. The cluster trying to share its jobs to its neighbors is known as 'power saver (PS)' cluster and the cluster that gets the share of the jobs of its neighbors is known as 'neighbor server (NS)' cluster. Here a single neighboring cluster is chosen to carry out the requests of the PS cluster because in this way the bandwidth cost can be reduced and the jobs can be served faster. As a result, the performance ratio increases as the delay for data transfer decreases. The performance ratio (PR) can be calculated as

$$PR = \frac{\delta_{tcppr}}{\sigma_{tcptar}}, \tag{1}$$

where, $\delta_{tcppr}$ and $\sigma_{tcptar}$ are the time of computation using predicted resources and total available resources, respectively.

In case of choosing the NS cluster, the PS cluster first consults with all of its neighbors to get their occupancy ratio,

$$\mu = \frac{\phi_{truc}}{\varphi_{tarc}}, \tag{2}$$

where, $\phi_{truc}$ and $\varphi_{tarc}$ are the total number of resource usage and allocatable resources in the cluster, respectively.

PS cluster will choose the NS cluster which has the minimum $\mu$ value from its neighboring NS clusters. Whenever the PS cluster migrates all its working VMs to the NS cluster, it can switch to sleep mode. As a result massive amount of energy can be saved and hence the cost of computation also decreases. Whenever a cluster goes to sleep mode of operation, if new request comes to it at that period, it simply forwards the request to the NS cluster. Whenever the value of the NS cluster rises above a predefined upper threshold level, it simply awakens the PS cluster to active mode and transfers the jobs it has received for the PS cluster to that PS cluster which is now operating in active mode. A NS cluster cannot go to sleep mode whenever it is serving the requests from some of its neighbors though its usage level goes below the lower threshold level.

In this section, we present our proposed intra- and inter-cluster scheduling algorithms for migrating virtual machines that allocate resources following distributed and energy-efficient approaches.

**Algorithm 2** *Intra-Cluster VM Scheduling Algorithm*

**INPUT**: *Resource information of each server*
**OUTPUT**: *Energy conserving intra cluster job scheduling*

1. **for** each server $i$ in the cluster **do**
2. Calculate $\omega_i$
3. **end for**
4. **for** each server $i$ in the cluster **do**
5. **if** $\omega_i < LWT$ **then**
6. **for** each server $j$ in the cluster **do**
7. Find highest $\omega_j$ such that $\omega_j + \omega_i <= UWT$
8. **end for**
9. Migrate VMs from server $i$ to server $j$
10. Keep server $i$ in sleep mode
11. **end if**
12. **end for**

### A. Proposed Intra-Cluster Scheduling Algorithm

For achieving computational benefits, we have considered all physical servers are of same capacity, i.e., all physical servers of same type have equal amount of total servable resources. As a result, in case of VM migration or transferring in service requests from one server to another within a cluster, the requirement of resources does not change and the new workload ratio for the servers can be computed by using simple addition and subtraction. Workload ratio ($\omega$) can be given as

$$Workload\ ratio(\omega) = \frac{\varsigma_{tu}}{\sigma_{ta}} \times 100\%, \tag{3}$$

where,
$\varsigma_{tu}$=total resource usage in the server
$\sigma_{ta}$=total allocatable resources in the server

Intra-cluster scheduling algorithm will be run by the Resource Allocator (RA) of a cluster. Whenever new requests come, RA assigns free VMs to the requests. During peak hours, resource utilization increases and hence the workload ratio ($\omega$) also increases. However, if the servers continue to provide the same amount of resources during off peak hour, huge amount of energy wastage will be occured. Hence RA will migrate VMs to a few working servers and keep the rest of the servers in sleep mode during off peak period to save energy.

At beginning, RA calculates the $\omega$ value for all the servers within the cluster. After that, RA will try to transfer all the requests under process from the servers with $\omega$ value lower than a predefined threshold, Lower Work Threshold (LWT), to servers having higher $\omega$ value and which can accommodate the whole work. Ther server, which serves the requests of both of the servers, should have $\omega$ value lower than Upper Work Threshold (UWT). Finally, RA will keep the released server in sleep mode to save more energy. The algorithm 2 presents the operation in detail.

### B. Proposed Inter-Cluster Scheduling Algorithm

Each cluster first calculates the occupancy ratio for itself. It also gets occupancy ratio from its neighbors to fill up the Neighborhood Occupancy Matrix (OMat). At any period of operation, whenever the occupancy ratio of a cluster gets down of lower threshold (LTH), it checks from its neighbors which have the minimum occupancy ratio. It then checks whether the occupancy ratio of the selected cluster is greater than the upper threshold (UTH). If it remains below UTH, then the PS cluster will migrate all its working VMs and coming requests to the selected NS cluster. Finally, it goes to sleep mode. The operation is presented in the algorithm 3.

**Algorithm 3** *Inter-Cluster VM Scheduling Algorithm*

**INPUT**:
$\mu$ : *Occupancy ratio*,
$OMat$ : *Neighborhood occupancy matrix*,
$N_{neighbor}$ : *Number of neighbors*
**OUTPUT**: *Energy conserving inter cluster job scheduling*

1. Calculate $\mu$
2. **if** $\mu < LTH$ **then**
3. **for** $i = 1$ to $N_{neighbor}$ **do**
4. Find the minimum $\mu_i$ from $OMat$
5. **end for**
6. **if** $(OMat[j] + \mu) < UTH$ **then**
7. Set $OMat[j] = OMat[j] + \mu$
8. Keep this cluster in sleep mode
9. **end if**
10. **end if**

### C. Energy Consumption

The energy model can be established based on the ongoing computation in each cluster. The energy optimization in a particular cluster can be given by

$$E_c(t) = (1 - \frac{\sum_{i=1}^{m} \omega i}{m \times 100}) \times E_{idle} + (n - m) \times E_{sleep}, \tag{4}$$

where, $E_c(t)$ is the amount of energy conservation for each cluster at any particular instance of time $t$, $n$ is the number of servers in the cluster and $m$ is the number of On (active + idle) servers in the cluster. $\frac{\sum_{i=1}^{m} \omega i}{m \times 100}$ gives the ratio of average workload in all the servers within a cluster. $E_{idle}$ is the amount of energy that can be saved when all the servers are in idle state. $E_{idle}$ multiplied by the server idleness ratio, gives energy saved by the portion of servers operating in idle state. (n-m) gives the number of servers operating in sleep mode. $E_{sleep}$ gives the energy saved by a server which is operating in sleep state. (n-m) multiplied by $E_{sleep}$ gives the energy that can be saved by servers in sleep mode. We can get the total energy conservation of a particular time period if we integrate the energy conservation within that period.

$$E_c = \int E_c(t) dt. \tag{5}$$

In terms of resource utilization the total amount of energy that will be consumed by the whole cloud data center can be given as

$$E_i = \alpha E_{i-1} + (1 - \alpha) \times E_{max} \times CCR, \tag{6}$$

where, $\alpha$ is the weight factor and $E_{i-1}$ is the energy consumption in the previous phase. $E_{max}$ is the maximum

amount of energy that could be consumed by the whole data center. $CCR$ is the computation cost ratio, which is the ratio of the computation cost of currently active servers to the computation cost that is incurred when all the servers in the cluster is in active state. The value of $CCR$ is taken at the phase of energy computation. Since $CCR$ is a fraction and value of $CCR$ is less than 1, Emax $\times$ $CCR$ gives the current energy consumption in the data center. That is Emax $\times$ $CCR$ gives a fraction of maximum amount of energy consumption in a data center which is used for utilizing computation resources in the data center at the time of computation. The $CCR$ can be given by

$$CCR = \sum_{i=1}^{N} \mu i. \tag{7}$$

The value of weight factor $\alpha$ can be adjusted with the dynamic nature of resource requirement and energy usage in the data centers. These equations for energy computation are used in the performance evaluation section to compare the result of our proposed VM scheduling algorithms with the existing state-of-the-art works.

## V. Performance Evaluation

In this section, we have evaluated the effectiveness of our proposed VM scheduling algorithm (VSA) using CloudSim[15]. We have compared performance of our VSA algorithm with that of Energy-aware Hierarchical Scheduling (EHS) [2] and Energy-aware Scheduling for Infrastructure Clouds (ESI) [4] in terms of energy consumption, number of servers in sleep mode and bandwidth utilization.

We have configured a data center of 5 clusters, each having 4 physical servers, where, each server has one quad-core processor, 8 GB RAM and 1 TB storage. Each core of a server is modeled with performance equivalent to 3000 MIPS. The cluster members are connected through 1 Gbps links and inter-cluster links are of 500 Mbps. Since all the hosts are of same capacity, we have considered LWT to be 15 percent usage of the total resources in a server and UWT to be 85 percent for all the servers. For the clusters, the value of LTH is considered to be 20 percent of usage of the total resources in the cluster and UTH is considered to be 90 percent of usage. The request arrival rate at the cloud data centers is randomly taken from the range of 30-120 requests per minute and the serving time of each request is taken from the range of 5-50 minutes.

In Figure 3(a), the relation between request arrival rate and the percentage of servers operating in sleep mode has been shown. The figure shows that, as the request arrival rate increases, the number of the VMs also increases. However, if we apply our proposed VSA, greater number of VMs can be kept unused hence energy consumption reduces. Again, some clusters can be kept in sleep mode at time of low request arrival rate and as a result, the total VMs as well as physical server at sleep state increases. Due to dynamic workload, VM re-scheduling and considering bandwidth optimization at the same time, better amount of energy can be saved by applying VSA compared to EHS and ESI. As the rate continues to increase, the difference in energy optimization degrades.

Figure 3(b) shows the level of energy that is consumed in the simulation environment by applying the three different VM scheduling model. The bar diagram shows that as the request arrival rate increases, energy consumption increases proportionally. Energy consumption level reveals the ratio of energy currently used to the highest amount of energy that can be used by the data center. This ratio is given here in percentage. Energy consumption can be calculated from the energy saving equations given before. The diagram shows the comparative study of energy consumption among the considered models.

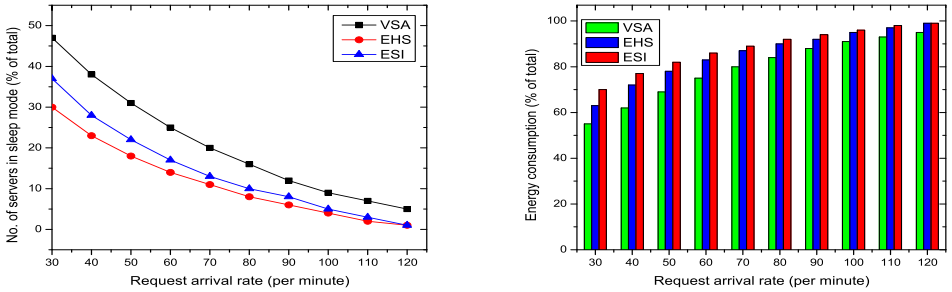
Figure 3(c) shows the energy optimization in case of workload reduction. Here the workload is shown as the total amount of resources used by VMs. It reflects the ratio of resources used by all VMs to the total amount of resources in the data center in percentage. As the workload reduces, the resource used by VMs also reduces since the number of VM reduces. Since we have used the inter and intra cluster algorithm to minimize the number of working VMs, our proposed algorithms are able to control the energy consumption in almost the same rate as the workload increases or decreases. Again we have considered VM migration at the time of serving requests not only at time of request placement at physical servers. Thus as the requirement reduces from peak hour to off-peak hour, our energy optimization algorithms perform accordingly. The results show that considerably big percentage of server nodes can be switched off by applying VSA compared to EHS and ESI.

In Figure 3(d), the bandwidth usage $B_{total}$ for different VM scheduling models is shown for different arrival rate of requests. Here bandwidth usage is given by the ratio of bandwidth used by VMs to the total bandwidth capacity in the data center. Bandwidth used by VMs can be calculated from the equations for bandwidth calculation given in previous section. The results show that bandwidth usage is also proportional to the request arrival rate and so is the cost for communication. However, because of considering communication cost into the scheduling of VMs, VSA and EHS show almost the same result. For ESI, bandwidth cost is greater than our proposed VSA for the same reason.

## VI. Conclusion

This work advances the cloud computing environment in several ways. $Firstly$, it plays a vital role for reducing data center energy consumption costs, which will help to develop a strong and competitive cloud industry throughout the world. $Secondly$, the consumers satisfaction will be increased through service level agreement assurances. $Thirdly$, we are able to use the maximum resource of every server, which leads us to obtain energy-efficient environment. We know that, large and rapidly growing big data centers in cloud are the significant source of $CO_2$ emission. Around the world, reducing greenhouse gas is the key concern and many countries treat this as a key energy policy. We have presented and simulated our algorithms in a data center cloud environment and the experiment results have shown that this approach can leads to a substantial reduction of energy consumption in data management and operation of data center. Simulation results show that, in terms of energy consumption and stability our approach outperform the others.

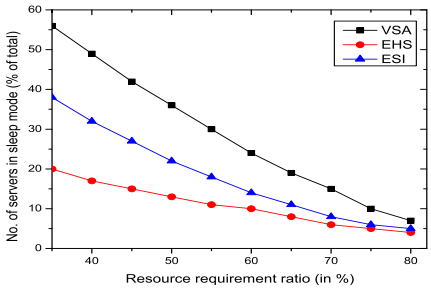
In future, we will include more energy-efficient factors in our algorithms for improving its further performance. For



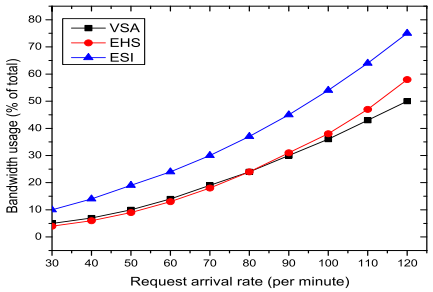

(a) No. of servers in sleep mode




(b) Energy consumption




(c) No. of servers in sleep mode




(d) Bandwidth Usage

Fig. 3. Performance comparisons

example, we will try to include the cooling condition of physical servers since it might put a good level of impacts on the performance of scheduling algorithms. Such improvements can introduce green ICT-based smart solutions for the next generation systems.

## ACKNOWLEDGMENT

This work is supported by the Information Society Innovation Fund (ISIF) Asia Project Grant 2013 (Driver Distraction Management Using Sensor Data Cloud). Dr. Md. Abdur Razzaque is the corresponding author of this work.

## REFERENCES

[1] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," in *Future Generation Computer System journal,*, Vol. 28, no. 5, May 2012, pp.755-768.

[2] Gaojin Wen, Jue Hong, Xu C., Balaji P, Shengzhong Feng, Pingchuang Jiang, "Energy-aware hierarchical scheduling of applications in large scale data centers," in *International Conference on Cloud and Service Computing (CSC),* 2011.

[3] Anne-Cecile Orgerie, Laurent Lefevre, Jean-Patrick Gelas, "Demystifying energy consumption in Grids and Clouds," in *GREENCOMP, pp.335–342, International Conference on Green Computing*, 2010.

[4] Thomas Knauth, Christof Fetzer, "Energy-aware Scheduling for Infrastructure Clouds," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 2012.

[5] A.T. Velte, "Chapter One: Cloud Computing Basics," in *Cloud Computing: A Practical Approach*, ed: McGraw-Hill, 2010, pp. 3–22.

[6] VMware, "VMware Distributed Power Management Concepts and Use," in *VMware Inc., Palo Alto, USA, Tech. Rep.* IN-073-PRD-01-01, 2009.

[7] R. Buyya, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges," in *Proc. of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, NV, USA, 2010, pp. 1–12.

[8] M. Maheswaran, S. Ali, H. Siegal, D. Hensgen, and R. Freund, "Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems," in *Proc. Heterogeneous Computing Workshop*, 1999, pp. 30-44.

[9] S. Kailasam, N. Gnanasambandam, J. Dharanipragada, and N. Sharma, "Optimizing service level agreements for autonomic cloud bursting schedulers," in *Proc. Intl. Conf. on Parallel Processing Workshops*, 2010, pp. 285-294.

[10] Ismael Solis Moreno, Jie Xu, "Customer-aware resource overallocation to improve energy efficiency in realtime Cloud Computing data centers," in *Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference*, 2011.

[11] O. Shy, "Overbooking, How to Price" in *Cambridge University Press*, 2001.

[12] K. Djemame and M. H Haji, "Grid Application Performance Prediction: a Case Study in BROADEN," in *presented at the First International Workshop on Verification and Evaluation of Computer and Communication Systems (VECoS 2007)*, 2007.

[13] Baliga, J.; Ayre, R.W.A.; Hinton, K.; Tucker, R.S, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," in *Proceedings of the IEEE*, 99(1), pp.149–167 (2011).

[14] Amit Kumar Das, Tamal Adhikary, Md. Abdur Razzaque, Choong Seon Hong, "An Intelligent Approach for Virtual Machine and QoS Provisioning in Cloud Computing," in *International Conference on Information Networking ICOIN, Bangkok, Thailand*, 27–30 January, 2013.

[15] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," in *Software: Practice and Experience, vol. 41, no. 1, pp. 23-50*,2011.