Journal of Computer and Knowledge Engineering, Vol. 1, No.2. 2017.
DOI: 10.22067/cke.v1i2.57729

Migration Management in Sensor-Cloud Networks

Farahnaz Farazestanian Seyed Amin Hosseini Seno*

Abstract: Placement of virtual sensors in servers of cloud environment is one of the most important issues in resource management of sensor-cloud networks. Virtual sensors allocate resources of cloud servers to themselves, run different applications and use equivalent physical sensors. Selecting an appropriate method for placement of virtual sensors on cloud servers has a significant impact on resource management and energy consumption. Also, because of the real-time requirements of sensor-cloud networks, traffic-aware placement is necessary to minimize delay in a network and respond to user requests in the shortest possible time. A new method for traffic-aware placement and migration of virtual sensors in sensor-cloud networks is proposed using the unique characteristics of sensor-cloud networks such as grouping virtual sensors and sharing them among different applications. This approach tries to meet the needs of users and the quality of service expected by the applications. Also, the resources of cloud servers are managed properly in this approach. The results of simulation show an optimization in energy consumption and also a reduction in traffic costs and reduction in service level agreement violation.

Keywords: Sensor-cloud networks, virtual sensors, resource management, migration, energy consumption, traffic.

1. Introduction

Sensor-cloud networks are proposed to overcome limitations of sensor networks and provide new services for users as a new generation of cloud computing. These networks benefit from cloud infrastructure for sharing physical sensors among different users. Also, some problems of sensor networks such as the limitation of data storage and the processing of data are eliminated [1-3].

Sharing physical sensors among different applications is performed by virtual sensors. Each virtual sensor is a software sensor, which is created as an intermediary for the connection between physical sensors and applications [4, 5]. Each virtual sensor is related to one or several physical sensors that can aggregate their data and perform some calculations on them. Virtual sensors can communicate with each other and form a virtual sensor group. It is also possible to share virtual sensors between different applications [6]. Virtual sensors allocate resources of cloud servers to themselves, including CPU, memory, storage, and bandwidth. They can respond to various requests of users. Virtualization technology enables the creation of virtual sensors in virtual machines of cloud servers. This technology uses different placement methods of virtual machines to manage allocation of cloud server's resources.

A review of existing literature in the field of sensor-cloud networks shows that the placement of virtual sensors on cloud servers has been neglected by the designers of these networks. In the available approaches, there is no mention of the impact of migration on meeting real-time needs of these networks. So, it seems that a new approach for migrating virtual sensors can have a significant impact on energy saving and network traffic and also meeting the quality of service expected by the applications.

2. Background

One of the challenging issues in data centers of clouds is the placement of virtual machines in cloud servers. Many approaches have been proposed to solve this problem. These approaches are meant to meet different objectives, such as increasing the efficiency of server resources, reduce energy consumption, reduce network traffic, or meet the quality of service (QoS) expected by the applications. However, it seems that this issue has not been addressed yet in sensor-cloud networks, and no method has been presented for optimization of this problem. What has been stated in sensor-cloud networks is a general case of creating virtual sensors on cloud servers [6]. In this case, after receiving requests of (from) users for using a special physical sensor, the sensor-cloud infrastructure retrieves the related template of that physical sensor from the repository. Then, it tries to provide the virtual sensor on the existing server. If the existing server is not capable of meeting the resource requirements of the virtual sensor, a new server is selected. In this paper, virtual sensor placement is not considered as an optimization problem, and there is no preference for selecting servers running virtual sensors. Also, in other research studies such as M. Yuriyama's work [3], general methods have been used and virtual sensor placement is not an independent issue. Therefore, the allocation of virtual machines in the cloud environment are studied in order to present a new method for placement of virtual sensors on sensor-cloud networks.

One of the proposed solutions for the virtual machine placement problem in a cloud environment is considering the available resources and capacities on cloud servers. Power consumption in the data center is related to processor, memory, storage and network interface. Among these, the processor has the highest proportion of power consumption. Therefore, optimizing the utilization of the processor is an important factor in the virtual machine placement problem [7]. The capacity limitations of the resources that are mentioned as server constraints, means that total resources allocated to all virtual machines running on a server cannot be more than the total capacity of that server [8, 9]. If the

Manuscript received July 24 , 2016; accepted. November 6, 2017.
F. Farazestanian, Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran, fa.farazestanian@stu.um.ac.ir.
*S. A. Hosseini Seno, Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran, hosseini@um.ac.ir.

constraints on a server are not met and the needed resources are not fully allocated to virtual sensors, virtual machine performance may be affected, and the expected quality of service is not achieved.

Migration that is one of the technologies in the cloud environment is the solution for distributing workloads on cloud servers. It helps avoid overloading the servers to maintain proper performance. Live migration of virtual machines is used to move virtual machines from one server to another during the running time and to respond to their workload changes [10, 11]. Xen is a virtual machine monitor (VMM) that manages and monitors the migrations of virtual machines on data centers [12]. This VMM is used in the simulation of the proposed algorithm as the hypervisor in cloud servers.

In multi-tier applications where each part of the application is executed on a separate virtual machine, dependencies between virtual machines are considered as an optimization parameter, which can lead to traffic on the network [8]. The algorithm in [13, 14] aims to migrate virtual machines so that network traffic is minimized and server-side constraints are satisfied. This algorithm considers a dependency graph between dependent virtual machines. The cost of migration is calculated as a function of traffic demand between each pair of dependent virtual machines and the distance between them. Distance is defined as the latency, delay or number of hops between each pair.

D. S. Dias et al. [15] have considered traffic dependencies between virtual machines. The algorithm tries to find dependent virtual machines by using the concept of graph community. In this method, virtual machines are considered as nodes of a graph and the dependencies between them are depicted by edges. The edge weight is used to determine the degree of relationship between a virtual machine and other virtual machines. This weight shows the level of traffic between each pair of virtual machines. Thus, with more dependency between virtual machines, a greater weight will be expected. After creating traffic matrix and identifying virtual machine communities [16], each community must be placed in one section of the servers called racks. Each rack contains a number of servers which are connected to one switch. The purpose is to place dependent virtual machines as close as possible to each other. The solution of this problem involves the amount of CPU and memory required by each community and the available resources in each rack. This problem is defined as a bin packing problem. H. T. Vu et Al. in [17] have tried to simultaneously reduce network traffic and energy consumption and increase the utilization of the processor by developing this algorithm. In this method, a tree is created such that in its lowest level, dependent virtual machines are placed as sibling nodes after finding dependent virtual machines. The host of the node which is selected for migration may be under-utilized or over-utilized. According to this, the destination is the host where the increase of energy consumption is minimum. Also the total distance between the destination host and dependent virtual machines should be minimum.

One of the proposed methods for making decisions about the start time of a migration is using thresholds according to CPU utilization. In the double-threshold method, the amount of CPU utilization is bounded between the upper and lower thresholds. Exceeding these amounts will force the server to execute migration [7]. If CPU utilization violates the low threshold, all the virtual machines on the server are forced to migrate and then the server will shut down. Also, if the CPU utilization exceeds the high threshold, the server should select one virtual machine for migration. Migrating the virtual machine will cause reduction in the CPU's workload. Consequently performance of the applications will also be managed in addition to reduction in energy consumption. Since the workload of applications is continuously changing, the CPU utilization also has continuous changes over time. So, the placement of virtual machines must be constantly optimized. Several heuristic algorithms have been proposed by A. Beloglazov et al. [18] with the ability to adapt to these changes. These algorithms are designed to optimize power consumption, but they do not consider traffic dependencies between virtual machines. In the method proposed in this study, these algorithms are used to select a virtual machine for migration.

Consolidation of virtual machines on a single server is also considered in some papers. Inappropriate consolidation of heterogeneous virtual machines on cloud servers will not only affects computing performance of applications, but it also reduces energy efficiency resulting in more energy waste [19]. Drop in efficiency also leads to violation of service level agreement (SLA). Placing one virtual machine on different servers can cause different energy efficiency levels due to the unique characteristics of the virtual machine. Because of the different effects of consolidations of virtual machines on a single server, implementing virtual machines to servers will have more complexity. Due to the characteristics of consolidated virtual machines, power consumption and overall efficiency will change. This is because of internal conflicts among consolidated virtual machines, such as cache contentions, conflicts at functional units of the CPU, disk scheduling conflicts, and network transfer conflicts. M. F. H. Bhuiyan et al. [20] have shown that a blind consolidation of virtual machines on a server will lead to reduced power consumption and energy loss. The idea of the researchers in [21] is placing virtual machines with the same dominant resources on different servers at the best effort. In this case, the resource competition between different virtual machines on the same server is significantly reduced. At the initial placement of virtual machines on servers, virtual machines are classified according to the dominant resource consumption. Then they are arranged in descending order according to their priorities. In each category, the virtual machine with the highest priority is assigned to the server with the greatest resource available of the category's type. Since the migration of virtual machines reduces efficiency, the machine that has a lower priority will be selected for migration.

In the placement of virtual machines in the cloud environment and the methods in the field of network traffic in the cloud, it is seen that the network traffic is considered between virtual machines and relevant applications or data transfer between data centers and cloud servers. However, in sensor-cloud networks the traffic can occur because of the characteristics of the network. The possibility of creating virtual sensors and virtual sensor groups has many benefits and simultaneously, the availability of new features for users of sensor-cloud networks, may cause unwanted effects on the performance of the network. When virtual sensors participate in a virtual sensor group, data communication among the members of that group will happen. According to the data collection rate of the associated physical sensors, the traffic load among the group members may change. In addition, if the locations of the servers of a group are far from each other, traffic between the servers will be imposed on network and traffic overhead can ensue. As a result, it may fail in responding to user requests and sending data to the applications. On the other hand, sharing virtual sensors between different applications will add the number of virtual sensor groups continuously, which increases the importance of this issue.

Energy consumption by servers of the cloud is also a significant problem. Many factors that should be considered can affect this and a suitable approach to optimize the energy consumption should be adopted. Placement of virtual sensors on cloud servers, as well as how to choose the host servers can be one of the factors. Also, using virtualization in cloud environments offers another feature called migration that can be used to perform this optimization.

Another challenging issue in cloud environments is resource management. Distribution of virtual machines on the cloud servers is one of the most complex issues in recent years, and various methods have been proposed to solve it. Meeting the quality of service expected by the applications depends on efficient allocation of resources needed by them in the running time. It has a significant impact on the quality of applications. Also, an inappropriate combination of virtual machines on a server can cause resource competition and reduce server performance. Because different combinations of virtual machines on a server have a great impact on energy consumption, the homogeneity of virtual machines on a server should be also considered in the placement process.

The main objective of the proposed method is introducing a traffic-aware algorithm for placement of virtual sensors by using live migration of virtual machines to distribute the workload. This is more important because of the grouping feature of virtual sensors. Resource management is taken into consideration with regard to energy efficiency. This approach should provide acceptable performance and hence it is expected that a service level agreement shall be fulfilled.

3. The proposed method

The method proposed for traffic-aware migration of virtual sensors in sensor-cloud networks in this study is looking for a suitable destination for migration. In this method, the virtual sensors are classified based on their dominant usage of resources, including processor, memory, and network bandwidth. When finding a suitable destination for migration, virtual sensors with the same dominant resource usage are distributed on different servers. Moreover, the destination of migration during a replacement of virtual sensors is selected according to virtual sensor groups. In this case, the destination server is selected as close as possible to the servers of dependent virtual sensors. Also, the destination server is selected based on the minimum energy increase after the placement. Assumptions necessary for the implementation of this method are described in section 4.1 and details of this method are explained in Section 4.2.

3-1. Assumptions and Limitations

In the proposed method, we assume that each physical sensor is associated with the cloud via the related virtual sensor. Each virtual sensor is running on a virtual machine on cloud servers. In this method, host servers of applications and host servers of virtual sensors are considered to be separate. The proposed algorithm tries to find an optimized allocation of virtual sensors on cloud servers and does not take into account the allocation of applications.

The proposed allocation method is designed by considering the two main characteristics of sensor-cloud networks. These features include the ability to create virtual sensor groups and meeting the needs of real-time data. Different types of data communication between the physical and virtual sensors are based on S. Madria's research [22, 23], including one-to-one, one-to-many and many-to-many communication between physical and virtual sensors. Based on the possibility of grouping virtual sensors, data communication and dependencies between virtual sensors of a group are used as parameters in the proposed method [6]. It is assumed that the virtual sensors will have no data communication with each other, except in the case of placement in a virtual sensor group.

In the proposed method, the data required by different applications are considered only in real-time situations, and the data stored in the data centers are not considered. Real-time requirements include traffic control to minimize the response time to the applications.

The initial placement of virtual sensors is based on M. Yuriyama's work [6]. When a client request is sent to the cloud environment for using a special physical sensor, the user will be assigned to that virtual sensor if a virtual sensor has already been created for that physical sensor. Otherwise, sensor-cloud infrastructure tries to create a new virtual sensor for each physical sensor based on predefined templates. To this, server resources should be reserved for the virtual machine which is running the virtual sensor. In this method, only three resource centers that include CPU, memory, and network bandwidth are exploited for each virtual sensor.

Each virtual sensor may dominantly benefit from one of the available resource centers. For example, a virtual sensor in a group may mostly need the processor to perform data integration and necessary calculations. Also, the virtual sensor whose equivalent physical sensor samples high volume data from the environment (e.g. video data) may need more memory. Physical sensors with a high sampling rate can also be an example of equivalent virtual sensors that are using a lot of bandwidth. Each virtual sensor is placed in one of the groups of dominant processor, memory, or network consumption, according to its dominant resource consumption. This classification can be obtained by monitoring the resource usage of virtual machines on a server [21]. According to the description of virtual sensor in S. Kabadayi's work [4, 5], the data type and sampling rate are determined by the user at the time of creating a new virtual sensor. These two parameters can also be used in the classification.

3-2. Migration of virtual machines

For optimization of the network status, cloud servers must use migration algorithms. Each migration algorithm should answer two questions: 1) which server should perform migration? And which virtual sensor should be migrated? In addition, 2) which server will be the destination of the migration? The main objective of the proposed algorithm here is to answer the second question. To address the first question, the algorithms presented by A. Beloglazov et el. [18] are used.

3-2-1. Selecting server and a virtual machine for migration

An important factor that affects the efficiency offered by cloud servers is the workload of virtual machines running on them. Among server resources, the CPU has the most significant impact on the performance of the virtual machines. If the workload on a server exceeds its acceptable level, the server will not be able to meet the needs of virtual machines and virtual machines will lose their quality. On the other hand, if the workload is too low on the server, the server resources will be wasted. This causes energy loss. So, optimizing CPU usage level is very important and effective, in this regard.

There are two general approaches to identify overloaded hosts: 1) Adaptive utilization threshold based algorithm, 2) non-threshold based algorithm. After detection of overloaded hosts, the virtual machine for migration should be selected. A number of algorithms for detecting overloaded servers and selecting the appropriate virtual machine for migration are proposed by A. Beloglazov et al. [18]. These algorithms belong to the second category. Based on the results of the simulations and evaluations that have been done in this current work, the LrMmt algorithm has the best result in energy savings and SLA. This algorithm includes two local regression algorithms (Lr), one used to select the overloaded host and another to calculate the minimum migration time (Mmt) for selecting the virtual machine for migration. Due to the desirable results produced by this algorithm, it is used in the proposed method in this study.

To identify under-loaded servers, a simple method is used. In this way, after finding overloaded servers and migration destinations, the remaining servers are arranged in order of CPU utilization and the server with the lowest utilization is selected. This server tries to migrate all virtual machines running on it, while the destination servers do not experience overloading. After migration is complete, the server is set to switch to sleep mode.

After detecting over loaded and under loaded servers, virtual machines running virtual sensors are selected for migration. The destination server of migration is selected based on the allocation algorithm that will be presented in the next section. The pseudo code of this method is shown in Algorithm 1.

Algorithm 1: Sensor-Cloud VM Selection Algorithm

- **Given:**

```
H              Available Host List
VMsToMigrate   VMs that are selected for migration
```

- **Algorithm:**

```
for each host h in H
    if isHostOverLoaded (LrAlgorithm, h)
        VMsToMigrate.add (getVMsToMigrateFromOverLoadedHost (MmtAlgorithm, h))
DoMigration (SensorCloudVMAllocationAlgorithm, VMsToMigrate)
VMsToMigrate.clear()
for each host h in H
    if isHostUnderLoaded (UnderLoadDetectionAlgorithm, h)
        VMsToMigrate.add (h.getVMs())
DoMigration (SensorCloudVMAllocationAlgorithm, VMsToMigrate)
```

3-2-2. Allocation of virtual machines

By selecting the appropriate virtual machine for migration, the destination server must be determined. To do this, the available resources of the selected destination server should meet the resource requirements of the virtual machine which is being migrated. Resource usage of virtual sensors depends on the type and amount of data collected by the associated physical sensor. The virtual sensors are classified based on their dominant usage of CPU, memory, and the network.

The main part of the proposed method for selecting the optimal destination considers two issues: the amount of traffic on the network caused by virtual sensor groups, and how to consolidate heterogeneous virtual sensors on a single server. In the following section each of these cases will be studied separately.

3-2-2-1. Destination host selection based on virtual sensor groups

The destination server should be selected so that the dependencies between dependent virtual sensors are considered and the traffic between them is reduced. Data communication between virtual sensors only observes in virtual sensor groups. If the virtual sensors of a group are far from each other, data transfer between them causes high traffic overhead on the network. Consequently, when placing virtual sensors of a group, the distance between them must be reduced as much as possible. To do this, an algorithm for detecting dependent virtual sensors is needed.

According to the method presented by H. T. Vu et al. [17], the traffic between virtual sensors is modeled by a complete graph. The virtual sensors are vertices and network communications are edges. Edge weights are the traffic weight between the virtual sensors. Then, lowest weight edges are removed recursively and this continues until dependent virtual sensor groups with the high amount of traffic remain as a set of isolated sub-graphs. From this graph a tree structure is created in which sibling nodes at lower levels are virtual sensors of a group (Figure 1). For selecting the destination server for migration, if the migrating virtual sensor has a sibling node in the tree, the location of the server is chosen such that the total distance of the server to the servers of dependent virtual sensors is minimum. In other words, the destination server must be the closest possible server to the servers of dependent virtual sensors.

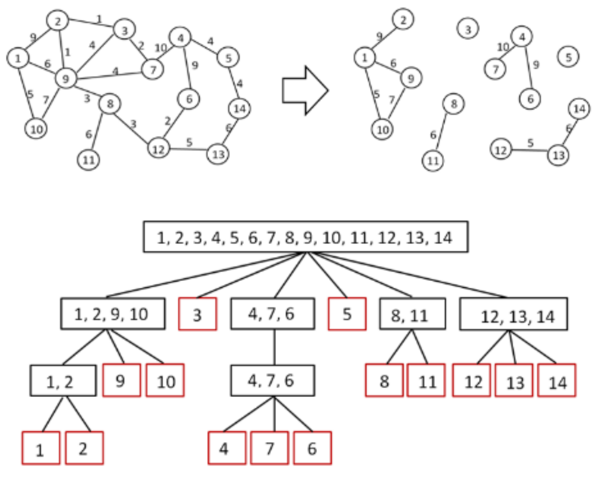

Fig 1. A conversion of graph to tree structure [17]

3-2-2-2. Destination host selection based on dominant resource consumption

As mentioned in Section 2, the consolidation of heterogeneous virtual machines on the same server has a high impact on performance and energy consumption. Replacement should be done such that the virtual sensors with the same dominant resource consumption are distributed on different servers as much as possible. In other words, these virtual sensors should not be placed on the same server. Consequently, conflicts and competitions for accessing the resources of servers will be reduced. By increasing the efficiency of virtual sensors, better service quality will be provided for applications. To achieve this, the destination server should be selected in such a way that the category of the migrating virtual machine's dominant consumption is different from the category of virtual machines' dominant consumption running on the destination server.

Moreover, for power management, the destination server should be selected such that the increase in energy consumption after the migration is minimum. The pseudo-code of the proposed allocation algorithm is represented in Algorithm 2.

Algorithm 2: Sensor-Cloud VM Allocation Algorithm

- **Given:**

```
H               Available Host List
V               migrating vm
allocatedHost   destination host for migrating  vm
```

- **Algorithm:**

```
for each host h in H
    if (h.consumptionType == v.consumptionType) then continue
    if ( h.isSuitableForVm(v) )
        if ( v.hasSibling() )
            if (h.TotalDistanceToSiblings( v.getSiblings() ) is minimum)
                allocatedHost = h
        else
            if (h.EnergyIncreseAfterAllocation() is minimum)
                allocatedHost = h
return h
```

4. Evaluation

For simulating the proposed algorithm and evaluating the energy consumption and traffic, the CloudSim toolkit [24] is used. It is an open source simulator for cloud computing environments, and it performs the modeling of virtual environments, on demand resource management and energy-aware simulations. Because of the absence of a suitable simulator for sensor-cloud networks, this tool is used.

The architecture considered for the data center is three-tier architecture. According to Cisco, today this architecture is mostly used in data centers in the cloud [25, 26]. Figure 2 illustrates this architecture. In the lowest layer of this architecture, the hosts are divided into partitions, and each partition is connected to an edge switch. Data communication between servers is via edge switches and data are not transferred to the higher layers. In the middle layer, each edge switch is connected to one or more aggregation switches. Then in the top layer, the core switches are connected to aggregation switches and also communicate with the rest of the data center. In the simulation of the proposed algorithm, this architecture is used for the data center of the cloud environment. The data center in the proposed data center includes 1 core switch, 3 aggregation switches, and 5 edge switches. Each edge switch is connected to 10 servers. Totally, the data center has 150 servers. The number of virtual machines equivalent to virtual sensors starts from 250 devices and in each round of simulation 10 devices are added to it. This amount goes up to 350 devices.

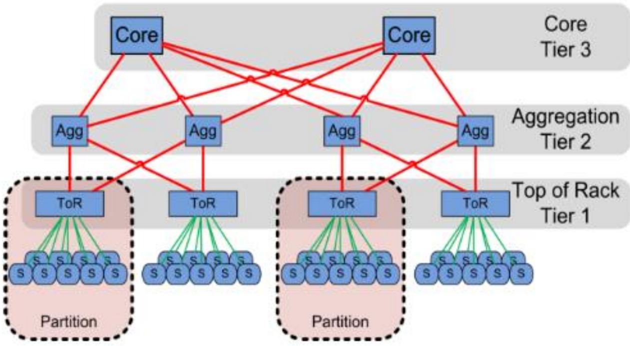

Fig 2. Three tier architecture for data centers [15]

The configuration of servers and virtual machines is according to the work of A. Beloglazov et al. [18]. The traffic in the data center is generated using FNSS tool. The run time of simulation is 20 minutes in each round. According to the generated traffic in the network, virtual machines are divided into three categories: CPU intensive, memory intensive, and network intensive. The servers are divided based on the number of virtual machines of each category running on them.

To select the appropriate virtual machine for migration, the algorithms presented by D. S. Dias et al. [15] are used. Since server selection and virtual sensors migration are done randomly in sensor-cloud networks, the random selection algorithm (LrRs) is used to simulate the migration in sensor-cloud networks [16]. Based on the results of the algorithms presented by D. S. Dias et al. [15], the LrMmt is nominated as the best algorithm for optimizing the energy consumption. This is combined with the proposed traffic-aware allocation method and is executed as the ScLrMmt algorithm. The evaluation parameters include: energy consumption, traffic costs, SLA violation, and the number of migrations.

4-1. Power consumption model

The main power consuming devices in data centers are processors, memory, hard disk, power generators and cooling systems. Recent studies [20] have shown that there is a linear relationship between the power consumption of servers and processor efficiency. Because of the complexity of power consumption modeling, especially in the new multi-core processors, instead of using an analytical model for the power consumption of the servers, the actual data released by the SPECpower benchmark is used for evaluating the power consumption level of servers.

The power consumption of the two servers, HP ProLiant G4 (http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110124-00338.html) and HP ProLiant G5 (http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110124-00339.html), based on the percentage of CPU utilization is shown in the Table 1. These two types of servers are used in the simulation here.

Table 1. Power consumption of servers based on CPU utilization

CPU Utilization	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4 Power Consumption (Watts)	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G4 Power Consumption (Watts)	93.7	97	101	105	110	116	121	125	129	133	135

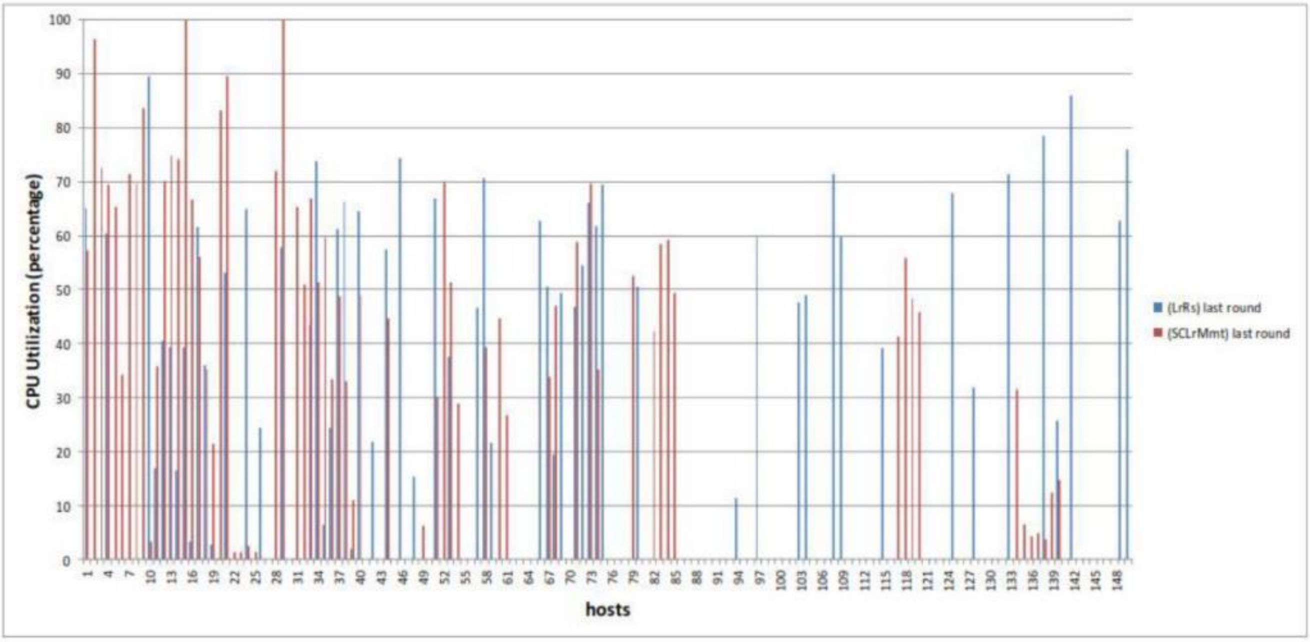

Fig 3. Average CPU utilization of servers

4-2. SLA Violation

Meeting QoS parameters is one of the most important issues in cloud environments. Requirements of quality of service are usually formulated as service level agreements and based on the characteristics like minimum throughput or maximum response time. Because in sensor-cloud networks, different applications use virtual sensors at the same time, SLA parameters must be defined independently from applications and their workloads. In assessment of the newly presented algorithm, SLA is calculated according to the P. Barham et al. proposed algorithm [18]. Thereby, the two parameters SLATAH and PDM are calculated first. Then, the following relationship is used to measure the service level agreement violation (SLAV) by combining these two parameters as follows:

SLAV=SLATAH . PDM

4-3. Simulation results

To evaluate the results of the proposed algorithm (SCLrMmt) and comparing it with the basic algorithm (LrRs), initially the impact of these algorithms on CPU utilization of cloud servers will be discussed. Among all the resource centers on a server, the CPU has a significant share of power consumption [18]. There is a direct relationship between CPU utilization and energy; i.e. a higher CPU utilization level causes more energy consumption. In addition, an increase in utilization leads to an increase in SLA violation since in higher utilization levels, virtual machine's access to the server processor will be restricted and competition for access will ensue. On the other hand, the power consumption of an idle server is about 70% of its power consumption in full usage state [27, 28]. Therefore, a suitable CPU utilization level is always desirable.

To check the CPU utilization of servers in both algorithms, simulations were run for 20 minutes with 150 servers and 350 virtual machines. The average of CPU utilization of servers during 12 rounds of simulation is illustrated in Figure 3. The proposed algorithm keeps the CPU utilization at a more appropriate level in comparison with the base algorithm. The average value of CPU utilization with the proposed algorithm was 27% while it was 25% for the base algorithm.

Figure 4 illustrates CPU utilization level at the end of the simulation. The average CPU utilization at this stage is about 46% in both algorithms. Although this amount is approximately equal in both algorithms, the proposed algorithm has tried to change the number of idle servers to off situation in order to reduce energy consumption. In the proposed algorithm about 50 servers have switched to off status during the simulation. This amount is about 20 in the base algorithm.

With changing the status of the servers, the virtual machines are forced to migrate and the new destination is selected based on the SCLrMmt algorithm. Initially, as is shown in Figure 5, the number of migrations (~320) is almost the same for both algorithms.

By increasing the number of virtual machines from 320, the SCLrMmt algorithm tries to turn off the idle servers and reduce their energy consumption. As a result, the number of migrations goes up to 390. While in the LrRs algorithm about 340 migrations are performed.

The comparison of energy consumption in the two algorithms is illustrated in Figure 6. As the number of virtual machines on the servers increases during the simulation, the energy consumption of servers also increases. Although the rate of increase in energy is almost the same in both algorithms, energy consumption in the case of using the proposed algorithm is lower. The SCLrMmt algorithm has about 7% more energy savings in comparison to the LrRs algorithm. This reduction in energy consumption is due to the shutdown of the idle servers and hence the resulting CPU usage optimization. Also, the appropriate placement of virtual sensors affects proper energy consumption.

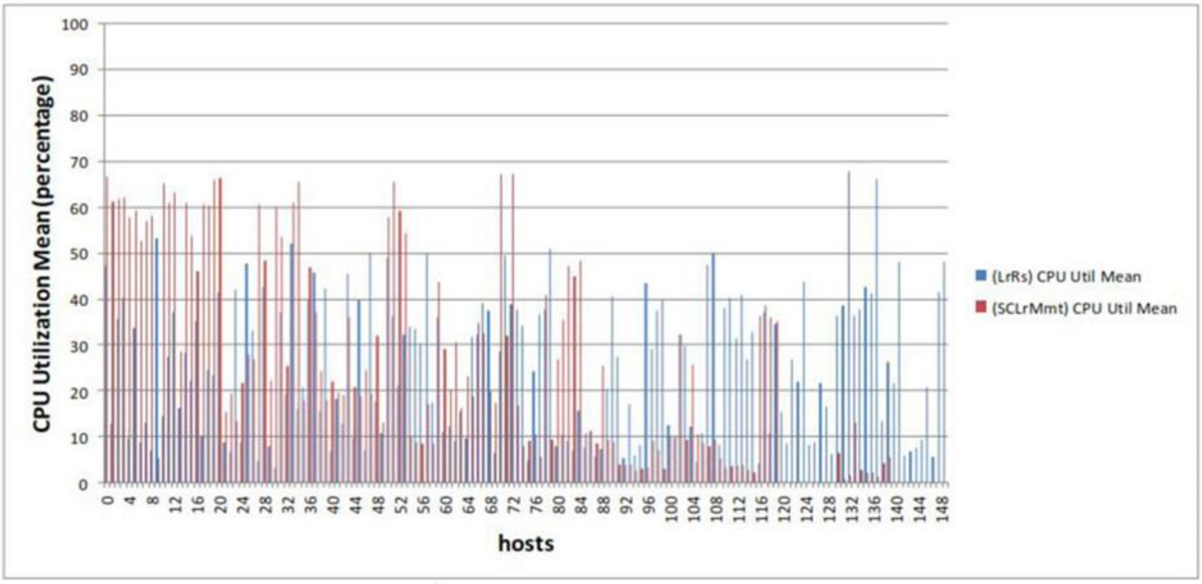

Fig 4. CPU utilization of servers at the end of simulation

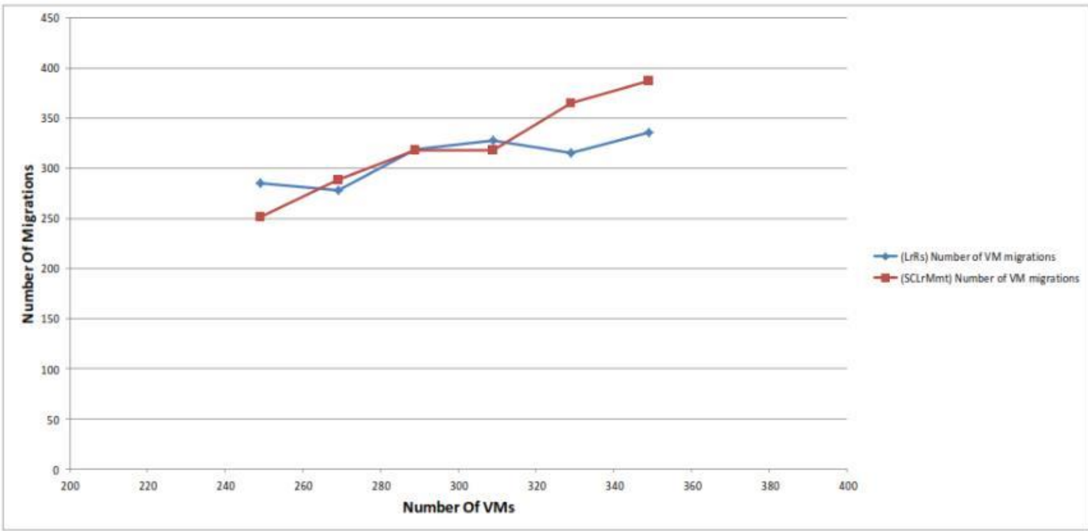

Fig 5. Number of migrations

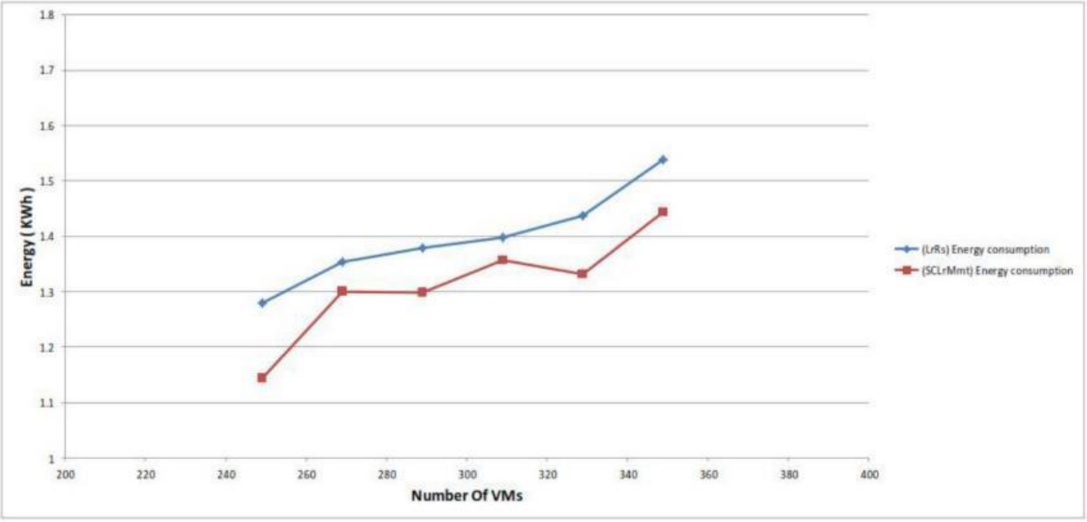

Fig 6. Energy consumption

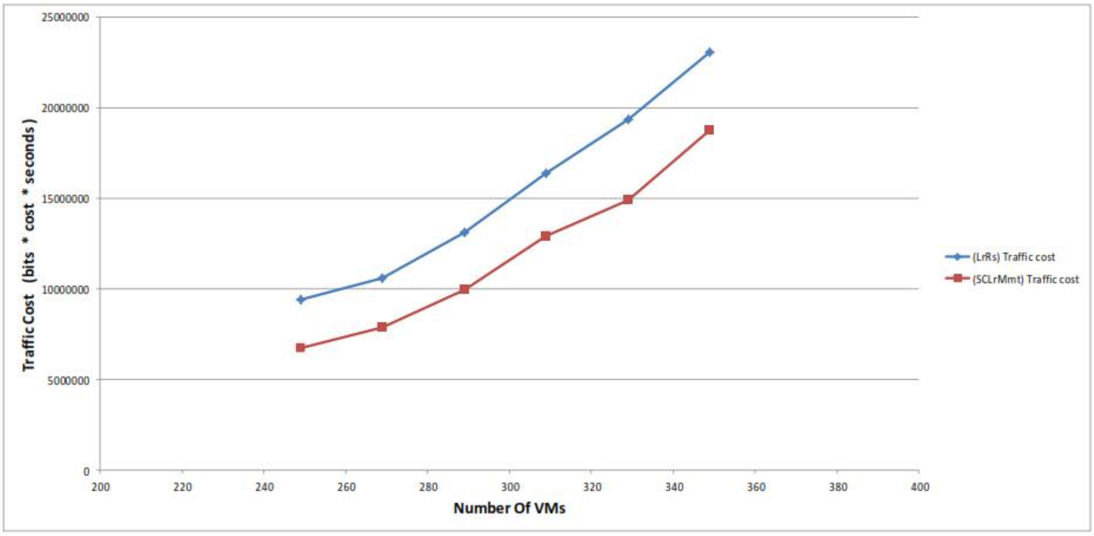

Fig 7. Traffic costs

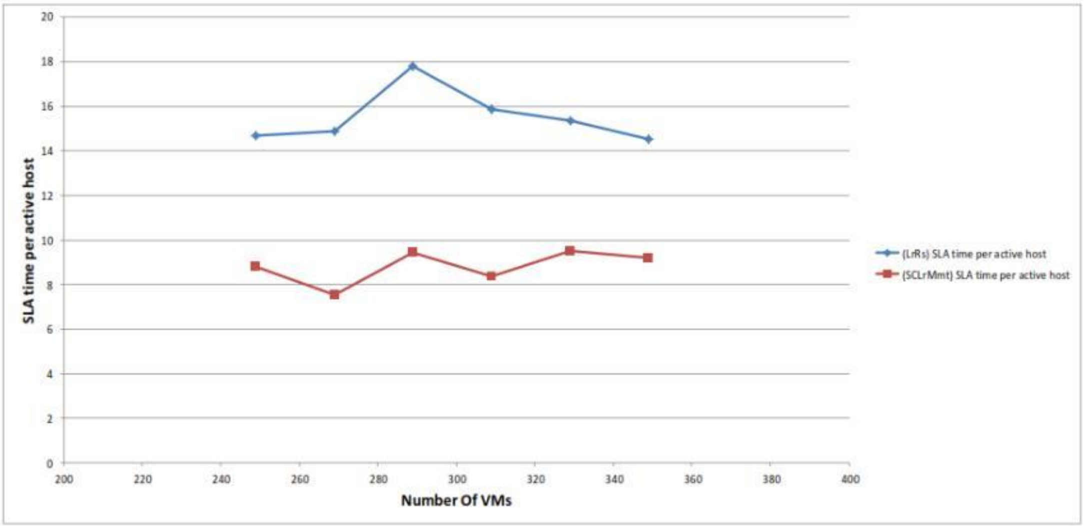

Fig 8. SLATAH comparison

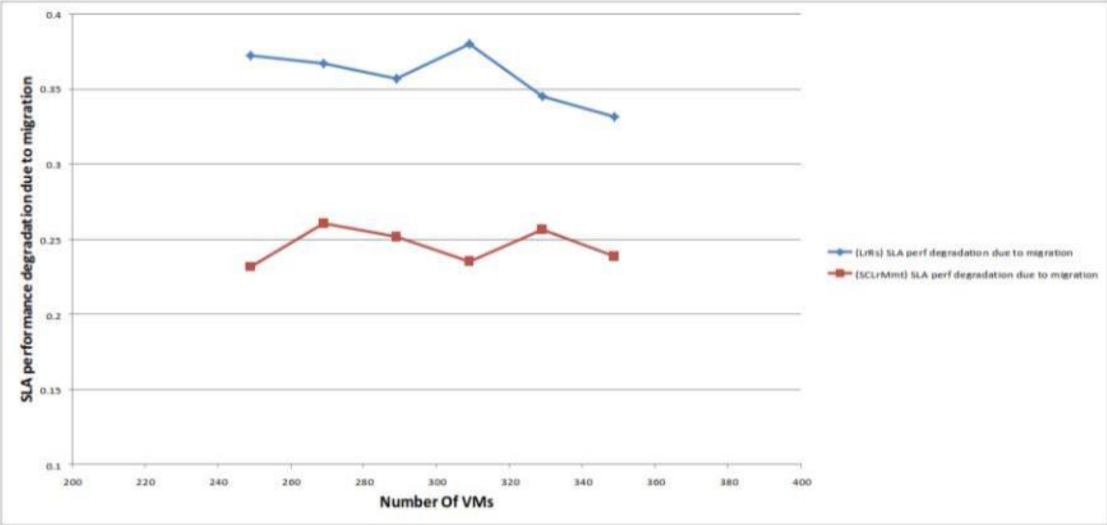

Fig 9. PDM comparison

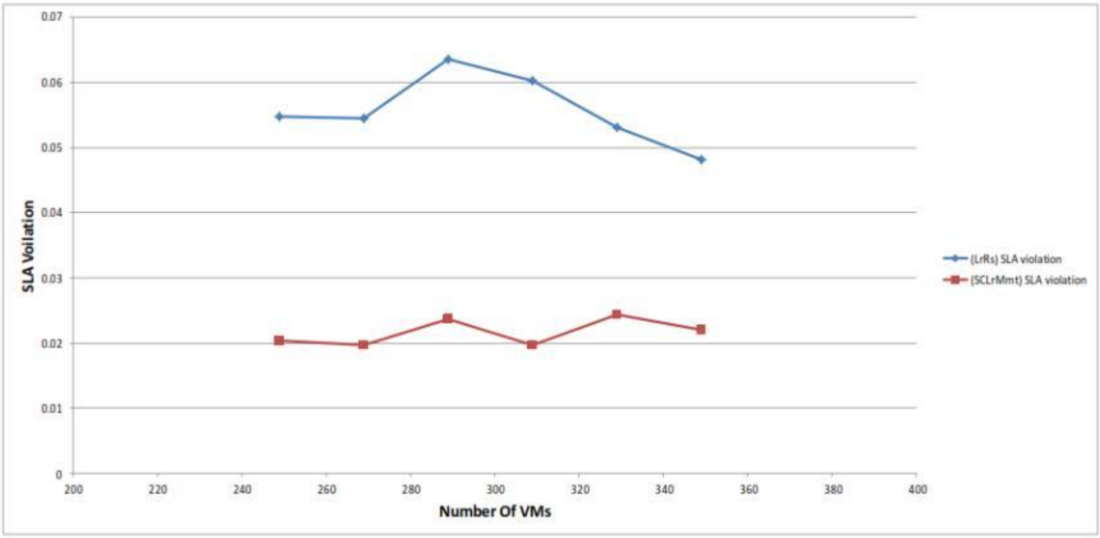

Fig 10. SLA violation

The traffic cost which is calculated based on the traffic between each pair of virtual sensors and the distance between them is compared for the algorithms in Figure 7. About 23% reduction in traffic cost is seen in the SCLrMmt algorithm. This is because the placement of virtual sensor groups are considered for choosing the destination of migration in order to minimizes the distance between the members of the groups. This type of placement results in reduction in traffic load and hence it lowers the cost.

The result of SLA violation for SCLrMmt and LrRs algorithm based on the two parameters SLATAH and PDM are illustrated separately in Figures 8 and 9. Subsequently, these two parameters are combined and SLA violation is calculated. Reduction in SLA violation is clearly visible in the proposed algorithm as shown in Figure 10.

5. Conclusion and future works

In this paper, a new approach for migration and reallocation of virtual sensors in sensor-cloud networks is proposed. By selecting the appropriate destination for migration, the proposed algorithm reduces the energy consumption in cloud servers. This method is based on minimization of the distance between the virtual sensor groups. It reduces the traffic overhead on the network and meets the real-time requirements of users. Reducing network traffic results in reducing violation of service level agreement. As a result, the desired quality of service is provided for applications. The results of the simulation of the proposed algorithm confirm improvements in energy efficiency, and decrease in traffic cost and SLA.

Since the main idea of the proposed algorithm for migration of virtual sensors is given from the migration of virtual machines in the cloud environment, there are similarities between them. Although in the new algorithm some special properties of sensor-cloud networks, such as the possibility of grouping and sharing virtual sensor groups are used, involving other parameters that are specific for sensor-cloud networks can clearly distinguish between cloud algorithms and sensor-cloud algorithms. Also, designing a tool that can simulate sensor-cloud networks influences the presentation of new algorithms. For example, just one virtual sensor is allocated to each virtual machine due to the lack of an appropriate simulator for sensor-cloud networks, while allocation of virtual sensors to a virtual machine can be a new idea in designing new algorithms. In the proposed algorithm, the parameters that decide on the start time of a migration are the overhead on servers and CPU utilization. Other factors such as the amount of memory can also have an effect on the occurrence of migration.

Future work on the proposed algorithm is based on the number of user requests for using one special virtual sensor. If the number of users of a virtual sensor is too high, it may cause increased delay in the response time to them. This situation is not suitable for real-time applications. Another issue for future work is considering data stored in data centers. In the proposed algorithm, only real-time data are considered and there is no control over the data stored in data centers. Minimizing the distance between virtual sensors and databases can have a significant impact on network traffic.

Reference

[1] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A Survey on Sensor-Cloud: Architecture, Applications, and Approaches", *International Journal of Distributed Sensor Networks,* vol. 2013, p. 18, Art. no. 917923, 2013.

[2] M. Körner, A. Stanik, O. Kao, M. Wallschläger, and S. Becker, "The ASCETiC Testbed - An Energy Efficient Cloud Computing Environment", *in Testbeds and Research Infrastructures for the Development of Networks and Communities: 11th International Conference,* TRIDENTCOM 2016, Hangzhou, China, June 14-15, 2016, Revised Selected Papers, S. Guo, G. Wei, Y. Xiang, X. Lin, and P. Lorenz, Eds. Cham: Springer International Publishing, pp. 93-102, 2016.

[3] Piraghaj, S. Fotuhi, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, A Survey and Taxonomy of Energy Efficient Resource Management Techniques in Platform as a Service Cloud. Handbook of Research on End-to-End Cloud Computing Architecture Design, 2016.

[4] S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: Abstracting data from physical sensors", *in In Proceedings of the International Symposium on on World of Wireless,* Mobile and Multimedia Networks, Buffalo-Niagara Falls, NY, 2006.

[5] S. Bose, N. Mukherjee, and S. Mistry, "Environment Monitoring in Smart Cities Using Virtual Sensors", *in 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud),* pp. 399-404, 2016.

[6] M. Yuriyama and T. Kushida, "Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing", *presented at the 13th International Conference on Network-Based Information Systems,* Takayama, 2010.

[7] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Generation Computer Systems,* vol. 28, no. 5, pp. 755-768, 2012.

[8] D. Huang, D. Yang, H. Zhang, and L. Wu, "Energy-aware virtual machine placement in data centers", *presented at the Global Communications Conference (GLOBECOM),* Anaheim, CA, 2012.

[9] F. Song, D. Huang, H. zhou, and I. You, "application-aware virtual machine placement in data centers", *presented at the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS),* 2012.

[10] C. Clark et al., "Live migration of virtual machines", *in Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation,* vol. 2, pp. 273-286: USENIX Association, 2005.

[11] U. Deshpande and K. Keahey, "Traffic-sensitive Live Migration of Virtual Machines", *Future Generation Computer Systems,* 2016.

[12] P. Barham et al., "Xen and the Art of Virtualization", *ACM SIGOPS Operating Systems Review,* vol. 37, no. 5, pp. 164-177, 2003.

[13] V. Shrivastava, P. Zerfos, K. W. Lee, H. Jamjoom, Y. H. Liu, and S. Banerjee, "Application-aware virtual machine migration in data centers", *in INFOCOM,* Shanghai, 2011.

[14] T. Chen, X. Gao, and G. Chen, "Optimized Virtual Machine Placement with Traffic-Aware Balancing in Data Center Networks", *Scientific Programming,* 2016.

[15] D. S. Dias and M. K. Costa, "Online traffic-aware virtual machine placement in data center networks", *presented at the Global Information Infrastructure and Networking Symposium (GIIS),* Choroni, 2012.

[16] J. Liu, J. Guo, and D. Ma, "Traffic Aware Virtual Machine Packing in Cloud Data Centers", *in 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity),* IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 256-261, 2016.

[17] H. T. Vu and S. Hwang, "A Traffic and Power-aware Algorithm for Virtual Machine Placement in Cloud Data Center", *International Journal of Grid & Distributed Computing,* vol. 7, no. 1, pp. Pages 21-32, 2014.

[18] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers", *Concurrency and Computation: Practice and Experience,* vol. 24, no. 13, pp. 1397-1420, 2012.

[19] M. F. H. Bhuiyan and C. Wang, "Energy-Efficient Virtual Machine Management in Heterogeneous Environment: Challenges, Approaches and

Opportunities", *presented at the International Conference on Systems, Man,* and Cybernetics (SMC), Manchester, 2013.

[20] M. Sharifi, H. Salimi, and M. Najafzadeh, "Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques", *The Journal of Supercomputing*, vol. 61, no. 1, pp. 46-66, 2012.

[21] X. Liu and L. Fan, "Priority-aware Gray-box Placement of Virtual Machines in Cloud Platforms", *presented at the arXiv preprint arXiv:1307*.6622, 2013.

[22] S. Madria, V. Kumar, and R. Dalvi, "Sensor cloud: A cloud of virtual sensors", *Software, IEEE,* vol. 31, no. 2, pp. 70-77, 2014.

[23] S. K. Madria, "Sensor Cloud: A Cloud of Sensor Networks", *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, 2016.

[24] R. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms", *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[25] "Cisco Data Center Infrastructure 2.5 Design Guide", May 2008.

[26] A. Gupta and N. Mukherjee, "Implementation of virtual sensors for building a sensor-cloud environment", *in 2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1-8, 2016.

[27] R. Buyya, A. Beloglazov, and J. H. Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges", CoRR, vol. abs/1006.0308, 2010.

[28] H. Goudarzi and M. Pedram, "Hierarchical SLA-Driven Resource Management for Peak Power-Aware and Energy-Efficient Operation of a Cloud Datacenter", *IEEE Transactions on Cloud Computing*, vol. 4, no. 2, pp. 222-236, 2016.