

Effective QoS-aware Novel Resource Allocation Model for Body Sensor-Integrated Cloud platform

Mohammad Mehedi Hassan, Majed Alrubaian, Atif Alamri

College of Computer and Information Sciences, King Saud University, Riyadh, KSA
mmhassan@ksu.edu.sa, malrubaian.c@ksu.edu.sa, atif@ksu.edu.sa

Abstract—This paper presents a cost-effective, scalable and energy-efficient cloud resource allocation model for real-time seamless access and processing of monitored body sensor data. In particular, two methods were used to solve the VM resource allocation problem. The first method uses a linear programming (LP) model of quantitatively optimizing VM allocation using cost objectives and constraints on the resource utilization condition, CPU utilization, energy consumption and delay of services. The second approach solves the VM resource allocation problem using different heuristics. We studied existing heuristics and modify them to fit our situation. We tested the workload of body sensor data on Amazon Cloud EC2. Then we create a large simulation scenario and compare our proposed method with other approaches.

Keywords— Body Sensor data, cloud computing, resource provisioning, and evaluation

I. INTRODUCTION

Over the past few years, the healthcare expenditure (e.g. pharmaceuticals, hospital services and medical devices) in the world has increased significantly [1]. It is due to the high prevalence of lifestyle diseases including cardiovascular diseases, diabetes, cancer and obesity. Currently, it is an important issue for any Govt. to look for economically sustainable healthcare programs. At the same time, rural and inaccessible areas suffer from a major problem in terms of high patient-to-doctor ratios. However, these two apparently dissimilar problems highlight a common and immediate need for an advanced, scalable and cost-effective pervasive healthcare platform that can benefit both the patients and health professionals by enabling real-time collection, dissemination and analysis of health data at anytime and anywhere.

Currently, there are many developments achieved in several technological and healthcare domains [2]. Body sensor networks (BSNs) are being used in healthcare domains for pervasive continuous monitoring of patients vital sign data for managing chronic conditions and detecting health emergencies. However, efficient management of the large number of monitored data is an important issue for its large scale adoption in pervasive healthcare services [3]. In this aspect, the integration of BSNs and cloud computing (bCloud) is becoming popular yet imposing several technical challenges such as how to design and develop a novel resource allocation model for the bCloud platform for real-time seamless access

and processing of monitored BSN data [5]. There are few studies [5-7] related to effective resource allocation to support various BSN services in cloud platform due to the dynamic nature of the BSNs and the range of physiological conditions. Moreover, processing health related media data like image, audio, video along with text data from BSNs require strict QoS guarantee. Therefore, it is not easy for a cloud provider to allocate resources for BSN services in advanced.

In this paper, we formulate the proposed VM resource allocation problem into the multidimensional bin-packing problem, which is NP-complete [5]. Similar to this problem, we consider each virtual machine as an item, and the dimensions as its capacities. The goal is to minimize the number of physical servers to be used to place all Virtual Machines, while considering physical servers' capacities. To reduce the response time, improve the overall resource utilization and avoid frequent VM migration, we additionally consider three constraints such as processing delay, overall resource utilization and special resource utilization. We further define a mechanism that dynamically determines the optimal threshold value of these constraints. Based on the above considerations, we design a MILP model for optimizing VM allocation into physical servers. We also use various heuristics to generate candidate resource allocation schemes and to choose the best scheme for real VM allocation. Several experiments were carried out to validate the efficiency of our proposed VM resource allocation model in bCloud platform.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 presents our VM resource allocation model and heuristics. Section 4 presents experimental results and performance comparisons. Finally, Section 5 concludes the paper.

II. RELATED WORKS

The integration of body sensor networks and cloud computing is getting much attention due to its immense potential. In [11] the authors proposed a framework to realize Healthcare as a Service (HaaS) model, wherein healthcare systems are defined as services in order to secure interoperability. By this model, a device or terminal is used to measure vital signs and transmits them to a virtual cloud environment. One can then check the patient's health status at anytime and anywhere by accessing the virtual server. In [12] the authors proposed a cloud-based system to automate the

process of collecting patients' vital data via a network of sensors connected to legacy medical devices, and to deliver the data to a medical centre's cloud for storage, processing, and distribution. In [13] the authors presented a Mobile Healthcare system prototype based on body sensor network and cloud computing for personalized healthcare monitoring and management. In [14] the authors reported a pervasive cloud initiative called Dhatri, which leveraged the power of cloud computing and wireless technologies to enable physicians to access patient health information at anytime from anywhere.

Many existing efforts [14- 18] study various VM resource management techniques for cloud resource management. However, few studies have been conducted related to effective resource allocation to support various BSN services in cloud platform. In [15] the authors proposed an autonomic resource provisioning framework to support real-time processing of vital signs and to acquire context awareness from BSN data. However, none of the above works so far presented provide a comprehensive model of a VM resource allocation for a bCloud platform that is scalable, dynamic, energy-efficient and cost effective.

III. PROPOSED RESOURCE ALLOCATION MODEL

At first we show the abstract system architecture of a general bCloud platform in Fig. 1. The system is comprised of five main components: body sensors, mobile device, cloud servers, users and display terminals such as Television (TV), personal computer or smart phones.

Such integrated system thus provide various BSN services to different users such as hospitals, clinics, researchers, or even patients ubiquitously by a variety of interfaces such as personal computer, TV and mobile phones. Now the basic concern of a VM allocation is that a physical machine must have enough capacity for hosting the VMs. To reduce the hosting cost, the number of active physical machines needs to be minimized. To save energy or power consumption, most power-efficient physical machines need to be selected. To avoid frequent VM migration, certain amount of CPU capacity needs to be preserved as backup resource for handling workload burst. To reduce the response time, the delay of the services needs to be controlled. According to above considerations, we used two methods which are described as follows:

A. Linear Programming Model

The input parameters and variables used in the linear programming formulation are presented in Table 1. For an atomic or composite media service I that needs to be allocated, the MILP model is presented in Eq. (1) to (6) as in [5]

$$\text{Minimize } \sum_{p \in P} y_p$$

Subject to

$$\sum_{p \in P} x_{pv} = 1 \quad \forall v \in V \quad (1)$$

$$\sum_{v \in V} u_{vr} x_{pv} \leq c_{pr} y_p \quad \forall p \in P, \forall r \in R \quad (2)$$

$$d_i \leq T \quad (3)$$

$$\sum_{v \in V} ruc_{pv} x_{pv} \leq T_1 \quad \forall p \in P \quad (4)$$

$$\sum_{v \in V} s_{pv} x_{pv} \leq T_2 \quad \forall p \in P \quad (5)$$

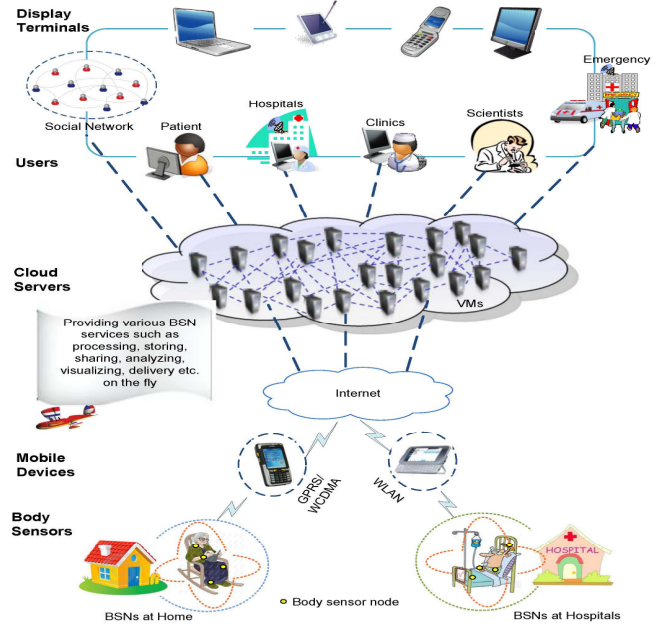


Figure 1. Abstract system architecture of a general bCloud platform

The objective function in (1) aims at minimizing the number of required physical servers. The constraint in (2) guarantees that each virtual machine is mapped to a single physical server. Equation (3) guarantees that the virtual machine demands allocated in each physical server do not overload its capacity. The constraint in (4) guarantees that the processing delay of media service I does not exceed a certain threshold value T . Equation (5) is an optional constraint which can help to improve the overall resource utilization. The optional constraint in (6) can reduce the chance of special resource overload and can potentially balance the special resource utilization among all physical servers. Here special resource means that some media applications may give more importance to only CPU than other resources or combinations of any resources.

TABLE 1. PARAMETERS AND VARIABLES FOR THE VM ALLCATION PROBLEM

Parameters	Description
P	set of physical servers
V	set of virtual machines
R	set of resources (CPU, memory, storage, GPU and network)
$I = \{v_1, v_2, \dots, v_n\}$	multimedia service (single or composite)
$u_{vr}, v \in R \ \& \ r \in R$	amount of resource r used by VM v
$c_{pr} \in R$	capacity for physical server $p \in P$ of resource $r \in R$
ruc_{pv}	overall resource utilization on physical server $p \in P$ after allocating virtual machine $v \in V$
S_{pv}	percentage of special resource utilization on physical server $p \in P$ after allocating virtual machine $v \in V$
$d_{p_1 p_2}$	delay between physical servers $p_1 \in P$ and $p_2 \in P$
d_{pe}	delay between physical server $p \in P$ and external server e
d_I	delay of media service I
$y_p \in \{0,1\}$	equals to 1 if physical server $p \in P$ is used, 0 otherwise
$x_{pv} \in \{0,1\}$	equals to 1 if virtual machine $v \in V$ is allocated to physical server $p \in P$ currently, 0 otherwise

In this scenario, the constraint in (6) can address to balance the special resource utilization throughout the physical servers. Since the future workload may not be predictable, our objective function in (1) represents the average statistics from time 0 to current time. For the constraints, they should be satisfied at any time when the allocation decision is made. Thus, we did not present the time variable t in those formulas. The optional constraints (5) and (6) are proposed to reduce the searching space for this NP-hard optimization problem. However, the use of these constraints may lead the results to be near-optimal. The definitions and effectiveness of ruc_{pv} , S_{pv} and d_I are explained as follows:

Definition 1: (media service delay): Given a multimedia service $I = \{v_1, v_2, \dots, v_n\}$, we discuss three types of service flow. The first case is for the atomic multimedia service case, where the VMs have no intercommunication with each other. In this case, the constraint on d_I can be defined as follows:

$$\sum_{p \in P} d_{pe_i} x_{pv_i} \leq T_{v_i} \quad \forall v_i \in I$$

We apply different threshold values for different Virtual Machines (T_{v_i} for v_i). For any virtual machine v_j that does not have communication with an external server, the corresponding constraint can be removed. The second scenario is for the composite multimedia service with synchronous integration, where the screen update from n Virtual Machines is synchronized first, and is then transmitted to the user. In this case, the constraint on d_I can be denoted as follows:

$$\sum_{p \in P} d_{pe_i} x_{pv_i} \leq \min(T_{v_i}) \quad \forall v_i \in I$$

Where $\min(T_{v_i})$ represents the most strict delay constraint among all of the services. Every virtual machine allocation should meet the requirement of $\min(T_{v_i})$, since they are synchronized.

The last case is for the composite multimedia service with sequential composition, where the output of the service running on v_i is the input of the service running on v_{i+1} . In this case, the constraint on d_I can be specified as follows:

$$\begin{aligned} \sum_{p \in P} d_{pe_1} x_{pv_1} &\leq T_{v_1} \\ \text{Given } p_1 \quad \sum_{p \in P} (d_{pe_2} + d_{p_1 p}) x_{pv_2} &\leq T_{v_2} - T_{v_1} \\ &\dots \\ \text{Given } p_{n-1} \quad \sum_{p \in P} (d_{pe_n} + d_{p_{n-1} p}) x_{pv_n} &\leq T_{v_n} - T_{v_{n-1}} \end{aligned}$$

In this case, the delay constraint on allocation is computed by subtracting T_v from T_{v+1} .

Definition 2 (Resource utilization and Special Resource condition): Given a media service or application i and a physical machine j , let c_{ij} , m_{ij} , g_{ij} , and b_{ij} are the percentages of resource usage regarding CPU, memory, GPU and network bandwidth, respectively. If the application is supported by the VM, then the extra resource consumption of VM, OS and remote server should be also included. Thus, application i and virtual machine v can be used interchangeably. Let fc_j , fm_j , fg_j , and fb_j be the percentages of idle CPU, memory, GPU and network bandwidth resources, respectively, on machine j . For any resource, if at least $k\%$ of free capacity is reserved to buffer the unexpected workload burst, then it would not be counted in the available idle resources. The exact amount of reserved resource is determined by the cloud resource provider by using long term benchmark or short term workload prediction models [32][33]. The media application can be allocated to

that physical machine only if the following condition (10) is met:

$$c_{ij} \leq fc_j \ \& \ m_{ij} \leq fm_j \ \& \ g_{ij} \leq fg_j \ \& \ b_{ij} \leq fb_j$$

After media application i is allocated on physical machine j , the average percentage of free resource ap_{ij} is defined using (11):

$$ap_{ij} = (fc'_j + fm'_j + fg'_j + fb'_j) / 4$$

where,

$$fc'_j = fc_j - c_{ij}, fm'_j = fm_j - m_{ij}, \\ fg'_j = fg_j - g_{ij}, fb'_j = fb_j - b_{ij}$$

For machine j , the resource utilization condition after allocating application i is denoted as ruc_{ij} , which is a mean-square value. Using (13), we have defined ruc_{ij} in (5) as follows:

$$ruc_{ij} = (fc'_j - ap_{ij})^2 + (fm'_j - ap_{ij})^2 + \\ (fg'_j - ap_{ij})^2 + (fb'_j - ap_{ij})^2$$

We assume that the applications running on a single physical machine share all of the resource capacity in a proportional way. More specifically, application i will be allocated on physical machine j , which will get $(c_{ij} / (1 - fc'_j)) \times 100\%$ of the CPU capacity, and same for memory, GPU and bandwidth resources.

As the resources may not be treated equally in multimedia system, constraint (6) is created to address this issue. Let us consider an instance where CPU is more important for processing the multimedia tasks, such as face recognition or data analytics. Regarding application i , the superiority of machine j is defined as s_{ij} in (6) by using (14):

$$s_{ij} = c_{ij} / (c_{ij} / (1 - fc'_j)) = 1 - fc'_j$$

It turns out that for allocation i the superiority of machine j is the percentage of CPU capacity that has been occupied after application i 's allocation. For other systems where bandwidth or GPU is more important, the definition of s_{ij} can be modified accordingly. In our MILP formulation, we have three types of threshold: network delay T in (4), overall resource condition T_1 in (5), and free resource T_2 in (6). The delay threshold T is given by the QoS requirement of each application. Certain amount of special resource capacity should be reserved to handle an unexpected resource burst.

Thus, T_2 will be generated according to the specific QoS requirement, the benchmark or the workload burst prediction regarding each application. The applications that experience frequent special resource bursts may require a small T_2 value.

B. Heuristics to Model the VM resource Allocation

The multidimensional bin-packing problem can also be solved using heuristics. Although heuristic solutions will not guarantee an optimal solution, the required time to obtain a feasible solution is much shorter than MILP. We utilize three heuristics like first-fit decreasing (FFD), best-fit decreasing (BFD), and worst fit-decreasing (WFD) and modify them according to the restrictions of the constraints (eq. (1) - (6)). In that heuristics, a lexicographic order is utilized to sort each VM demand. Following the heuristic definitions, the mapping of each VM will then be performed. In the FFD heuristic, the VM will be mapped to the first physical server with available capacity. In case of a BFD heuristic, it will be mapped to the physical server that leaves the least left over space after the mapping between all available physical servers. And for WFD heuristic, it will be mapped to the physical server that leaves the largest left-over space after the mapping between all available physical servers. We adopt these three heuristics to generate candidate VM allocation schemes, and to choose the best scheme for real VM allocation.

IV. EXPERIMENTAL RESULTS

In this section, we have presented simulation setup description and conducted several experiments to validate the efficiency of our proposed VM allocation approach as done in. We compared our proposed algorithm with three existing algorithms: a load balancing model, a queuing model, and a round-robin allocation [5]. The results include the performance of cost reduction; response time reduction and QoS guarantee. Table 2 shows simulation parameters, where HI, ACU, AMU, APU, ANU represents heterogeneity index, average CPU utilization(%), average memory utilization(%), average GPU utilization(%), and the average network bandwidth utilization(%) respectively. HI, ACU and AMU were retrieved from the Technical University of Berlin (TU-Berlin) workload [8], which are normally used by researchers and students to execute computational experiments. Table 3 specifies the delay settings, where IDC, SDC, IDT and SDT represents the individual delay constraint on atomic BSN service, the sequential delay constraint on adjacent BSN services, the individual delay time on a single server, and the sequential delay time on connected servers respectively.

TABLE 2. DETAILS OF WORKLOAD GROUP

Heterogeneity	Number of traces	HI	ACU	AMU	AGU	ANU
Low	50-200	0.17	25.2%	28.36%	30%	30%
High	50-200	0.63	47.28%	48.67%	50%	50%

TABLE 3. DETAILS OF DELAY

Heterogeneity	IDC	SDC	IDT	SDT
Low	25ms-35ms	25ms-35ms	5ms-30ms	5ms-15ms
High	15ms-45ms	25ms-45ms	5ms-30ms	5ms-15ms

While allocating the services, two types of allocation schemes are adopted. In large group allocation case, all BSN service requirements are assumed to be generated and submitted at one time. On the other hand, small group allocation allows only 1-5 media service requests submission. When the current allocation is done, the following 1-5 media service requests will be created. In both cases, each group of composite media services contains 1-5 services, which can be atomic, synchronous or sequential.

A. Cost Optimization

Several sets of experiments were conducted in the simulation. Firstly, we adopted different request patterns of media services/applications in the experiment (i.e. large/small media service group at Low/high heterogeneous environment) to measure the cost optimization capability. The number of media requests was fixed to 100.

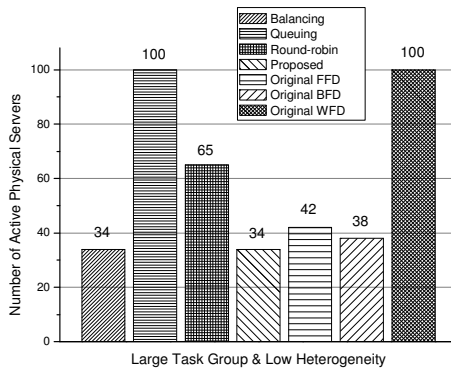


Figure 2. Cost optimization in small task group at low heterogeneity environment

Fig. 2 shows the simulation results derived from the large task group at low heterogeneity environment. From the results, we have found that our proposed approach and the load balancing method have the same performance. The optimal allocation sequence is applied by the load balancing method as well as our proposed approach. The queuing model performs worse than any other solution as it does not consider virtual machine. The round-robin method randomly chooses a physical server for each request. Since it does not provide any optimization method, we have found that 65 physical servers need to be launched according to the allocation results of round-robin method.

It can be seen from Fig. 4 that our proposed also achieves best performance in the small task group and low heterogeneity environment. The advantage of our proposed approach was greatly amplified in the high heterogeneity environment, which is more similar to the real scenario. The results are presented in Fig. 5. Compared with the load

balancing method, more than a 10% improvement is achieved from using our proposed approach. Thus, we can conclude that our proposed approach is more suitable for the real multimedia service allocation.

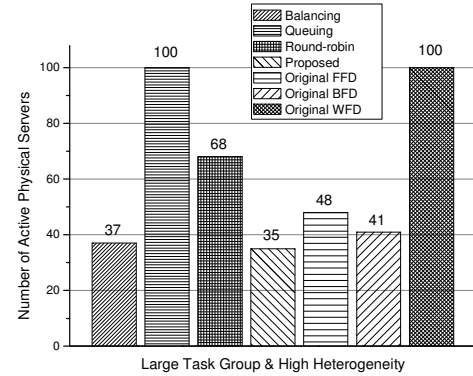


Figure 3. Cost optimization in small task group at high heterogeneity environment

In Fig. 3, we present the results retrieved from the large task group and high heterogeneity environment. Due to the resource utilization condition threshold, our proposed approach outperforms existing algorithms by avoiding overuse of any resource.

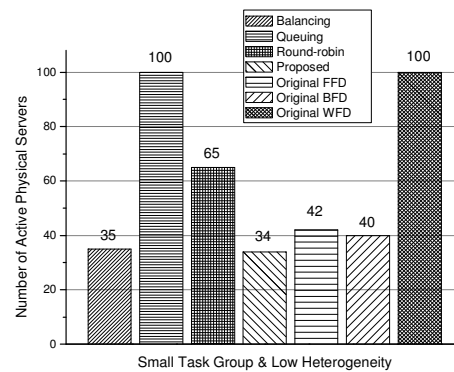


Figure 4. Cost optimization in small task group at low heterogeneity environment

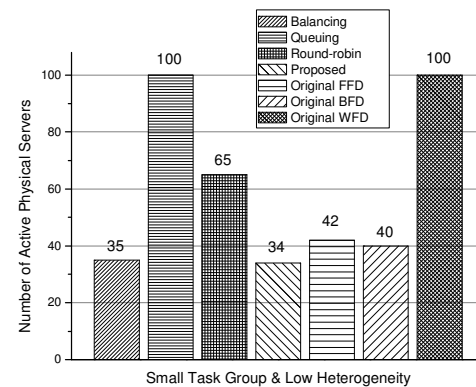


Figure 5. Cost optimization in small task group at high heterogeneity environment

B. Scalability of the Proposed Approach

Fig. 6 shows the results of the scalability test. When the total number of service requests equal 50, the performance of our proposed approach was slightly better than that of the load balancing method. By increasing the workload, a significant difference can be found in the graph. Compared with the load balancing method, our proposed approach demonstrates better scalability. The queuing method and the round-robin method have obvious drawbacks regarding scalability.

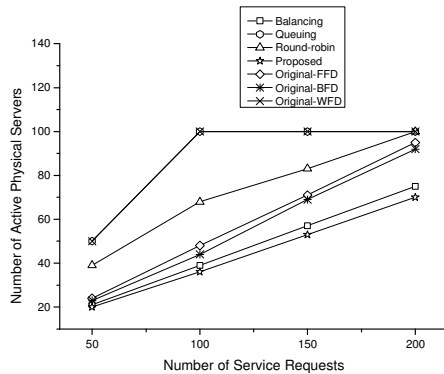


Figure 6. Scalability Test Results

V. CONCLUSIONS

This paper presents a VM resource allocation model that dynamically utilizes VM resources to satisfy QoS requirements of bCloud services or applications. In order to do VM resource allocation effectively, we have presented a MILP model, as well as heuristics. Performance comparisons show that our resource management/allocation approach performs very competitively while satisfying users' QoS demand. As for the future works, we would incorporate internal VM resource demand as a part of the future work. We believe that our proposed allocation approach can adapt such settings.

ACKNOWLEDGEMENT

This work was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (12-INF2885-02).

REFERENCES

- [1] E. Jovanov and A. Milenkovic. Body Area Networks for Ubiquitous Healthcare Applications: Opportunities and Challenges, *Journal of Medical System*, 35:1245–1254, 2011.
- [2] H. Viswanathan, B. Chen, and D. Pompili. Research challenges in computation, communication, and context awareness for ubiquitous healthcare. *IEEE Communications Magazine*, 50(5): 92-99, May 2012
- [3] M. C. Domingo. A Context-Aware Service architecture for the Integration of Body Sensor Networks and Social Networks through the IP Multimedia Subsystem, *IEEE Communications Magazine*, 49: 102-108, 2012
- [4] T. Gao, D. Greenspan, M. Welsh, R. R. Juang, and A. Alm, Vital signs monitoring and patient tracking over a wireless network," in the proc. of 27th Annual International Conference in Engineering in Medicine and Biology Society Shanghai, pp. 102-105, 2005.
- [5] M. M. Hassan, M. S. Hossain, A. M. J. Sarkar and E-N. Huh. Cooperative game-based distributed resource allocation in horizontal

- dynamic cloud federation platform., *Information System Frontiers*, Springer, Online First, 2012, DOI: 10.1007/s10796-012-9357-x
- [6] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput.Syst.*, 25(6), 2009
- [7] A. M. Kuo. Opportunities and Challenges of Cloud Computing to Improve Health Care Services, *Journal of Medical Internet Research*, Jul-Sep; 13(3): e67, 2011
- [8] C. Chatman. How cloud computing is changing the face of health care information technology. *J Health Care Compliance*. 2010 Jun;12(3):37–70.
- [9] A. Beloglazov, J. Abawajy, and R. Buyya, "Energyaware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, 2011.
- [10] I. Goiri, J. L. Berral, J. O. Fit, F. Juli, R. Nou, J. Guitart, R. Gavald, and J. Torres, "Energy-efficient and multifaceted resource management for profit-driven virtualized data centers," *Future Generation Computer Systems*, 2012.
- [11] M. Stillwell, D. Schanzenbach, F. Vivien, and H. Casanova, "Resource allocation algorithms for virtualized service hosting platforms," *Journal of Parallel and Distributed Computing*, vol. 70, no. 9, pp. 962 – 974, 2010.
- [12] F. Aisopos, K. Tserpes, and T. Varvarigou, "Resource management in software as a service using the knapsack problem model," *International Journal of Production Economics*, 2011.
- [13] H. Nguyen Van, F. Dang Tran, and J.-M. Menaud, "Autonomic virtual resource management for service hosting platforms," in *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, ser. CLOUD '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–8
- [14] W. Lin and D. Qi, "Research on resource self-organizing model for cloud computing," in *Internet Technology and Applications, 2010 International Conference on*, aug. 2010, pp. 1 –5.
- [15] F. Teng, Magoule, and s. F., "Resource pricing and equilibrium allocation policy in cloud computing," in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, 29 2010-july 1 2010, pp. 195 –202.
- [16] T. C. Ferreto, M. A. S. Netto, R. N. Calheiros, and C. A. F. De Rose, "Server consolidation with migration control for virtualized data centers," *Future Gener. Comput. Syst.*, vol. 27, pp. 1027–1034, October 2011.
- [17] M. M. Hassan, Song, B., Almogren, A., Hossain, M. S., Alamri, A., Alnuem, M. Hossain, M. A.. Efficient Virtual Machine Resource Management for Media Cloud Computing. *KSII Transactions on Internet and Information Systems (TIIS)*, 8(5), 1567-1587, 2014
- [18] G. Wei, V. A. V., Z. Yao, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *J. Supercomput.*, vol. 54, pp. 252–269, November 2010.

Mohammad Mehedi Hassan (M'12) is an Assistant Professor of Information Systems Department in the College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia. He received his Ph.D. degree in Computer Engineering from Kyung Hee University, South Korea in February 2011. His research interests include cloud collaboration, multimedia cloud, sensor-cloud, mobile cloud, thin-client, grid computing, IPTV, virtual network, sensor network, and publish/subscribe system.

Majed Alrubaian (M'15) is currently PhD student in the Department of Information Systems in the College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Kingdom of Saudi Arabia. His research interests include mining social data, cloud computing and sensor network.

Atif Alamri (M'12) is an Associate Professor of Information Systems Department, at the College of Computer and Information Sciences, King Saud University. Riyadh, Saudi Arabia. His research interest includes multimedia assisted health systems, ambient intelligence, and service-oriented architecture.