IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 6, 2017, accepted June 23, 2017, date of publication July 11, 2017, date of current version July 31, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2726012

Resource Provisioning for Cloud-Assisted Body Area Network in a Smart Home Environment

MOHAMMAD MEHEDI HASSAN[1], (Member, IEEE), HANOUF ALBAKR[1], HMOOD AL-DOSSARI[1], AND AMR MOHAMED[2], (Senior Member, IEEE)

[1]Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[2]Department of Computer Science and Engineering, Qatar University, Doha, Qatar

Corresponding author: Mohammad Mehedi Hassan (mmhassan@ksu.edu.sa)

This work was supported by Kind Saud University and Qatar University through the GCC Co-Funding Program under Grant GCC-2017-009.

ABSTRACT In recent years, cloud-assisted body area network (CABAN) technologies have made their entrance in the Smart healthcare field, such as Smart home environment, and play a significant role for healthcare data storage, processing, and efficient decision making. However, currently, the CABAN paradigm in the healthcare domain is facing increasing difficulty in handling the huge amount of sensor data that the body sensor devices generate from diverse Smart home applications. Therefore, the challenging is now timely storing, processing, and analyzing of the sensor data in real time to maintain the Quality of Service (QoS) requirements of the caregivers or Smart home applications. QoS, here, is the capacity to support diverse Smart home applications in healthcare with different priorities, performance, and resource requirements. Therefore, in this paper, we present a fast and robust cloud resource allocation model for body sensor devices to ensure QoS for Smart home healthcare applications. We develop the proposed resource allocation algorithm using agent-based modeling (ABM) and ontology. There are few works, which consider ABM and ontology for resource allocation in CABAN platform. Moreover, we used an ABM tool called NetLogo to implement the proposed resource allocation model. The results from the implementation were compared with the results of existing algorithms and found to be promising.

INDEX TERMS Cloud-assisted body area network, QoS, resource allocation, agent-based modeling and ontology.

I. INTRODUCTION

Due to recent developments of several technological advances such as body area network, Internet-of-Things (IoT), etc., smart homes [1]–[3] are becoming an effective solution to enable the growing number of elderly as well as physically impaired people such as deaf, dumb and blind people, to remain in their own homes with full support of lifestyles and care. With several body sensors based on IoT technology implemented throughout the home, as well as wearable health trackers, the health conditions of the residents are continuously monitored, and required assistance is triggered when an unusual or critical situation is detected. However, a smart home requires various technological solutions that should be suitable for their current physical conditions and situations. Although smart homes were designed more than a decade ago, there are several technical challenges still exist in these areas in terms of communication, sensing, health monitoring and home operations [4]–[8].

Currently, cloud-assisted body area network (CABAN) is becoming a promising technology in a Smart home environment to provide a powerful and scalable high-performance computing and massive storage infrastructure for real-time processing and storing of the body sensor data as well as analysis of the processed information under context to extract knowledge of the health condition of patients [9]–[12]. It can solve most of the body area network issues such as limited memory, energy, computation, and communication capabilities and also provide additional features such as ease-of-access, ease of- use, and reduced deployment costs and allows authorized persons to share and contribute, to follow work of colleagues, and to keep up to date with critical information in real time to improve decision-making.

The CABAN platform employs various IoT-based body sensors to collect comprehensive physiological information and uses gateways and the cloud to analyze and store the information and then send the analyzed data wirelessly to caregivers for further analysis and review [13]–[15]. Thus, CABAN can replace the process of having a health professional come by at regular intervals to check the patients' vital signs at home, instead providing a continuous automated flow of information. In this way, it simultaneously improves the quality of care through constant attention and lowers the cost of care by eliminating the need for a caregiver to actively engage in data collection and analysis [16]–[18].

Nevertheless, the research regarding the CABAN integration for Smart home platform is still in its infancy, and several technical challenges remain to be addressed to maximize the opportunities. One of the major challenges is how to timely storing, processing and analyzing of the huge amount of body sensor data in real time to maintain the Quality of Service (QoS) requirements of the users or pervasive healthcare applications. QoS requirements include low latency and network resource consumption, optimal utilization of computational recourses, etc. In CABAN scenario, QoS is the capacity to support diverse healthcare applications or workloads with different priorities, performance and resource requirements. A dynamic and fast resource scheduling for heterogeneous workloads in CABAN is critical for ensuring QoS.

In this paper, we present the development of a cloud resource allocation model for body sensor devices to ensure QoS in this framework, which has traditionally proven as NP-hard problem due to various constraints. Our goal is to provide the right resources (computing, storage, etc.) to the right persons at right time and at the right place. We proposed a fast and robust resource allocation algorithm by using Agent-Based Modeling (ABM) and ontology [20]–[22]. In recent years, Agent-based modeling is becoming an effective method to solve such complex problems. We combine ontology with ABM and show the relation between them. There are few works, which consider ABM and ontology for resource allocation in CABAN platform. Furthermore, we optimize the resource allocation model using Mixed Integer Linear Programming (MILP) to find the most suitable servers for VM allocation. The important agents of the proposed algorithm, the architecture of the proposed solution and the detailed flowchart of the proposed algorithm are also provided. The results from the implementation are compared to the results of existing algorithms and found to be promising. The stability analysis to the implementation exhibits the robustness of the algorithm and ABM. These results motivate future research of using ABM as a tool for solving complex problems within the CABAN framework.

The reset of the paper is organized as follows: Section 2 describes the related works. Section 3 introduces the proposed resource allocation model for CABAN platform. Section 4 describes the proposed optimization model for resource allocation. Experimental evaluation presented in Section 5. Finally, Section 6 concludes the paper.

II. RELATED WORKS

A. GENERAL RESOURCE ALLOCATION ALGORITHMS IN CABAN HEALTHCARE ENVIRONMENT

Few studies have been conducted related to effective resource allocation to support heterogeneous body area network or Internet of Tings (IoT) tasks in CABAN system along with ensuring quality of services such as real-time processing, storing, sharing, prioritizing, visualizing, and analysis of monitored data as well as acquiring context-awareness. Resource allocation is challenging in a CABAN healthcare environment due to the diversity of context-aware environments, the range of physiological conditions and the dynamic nature of the resource constraint IoTs [23], [24]. Moreover, the CABAN platform brings health-related media data like image, audio, video along with text data, which require strict QoS guarantee. It is cumbersome for a cloud provider to perform over commitment of VM resources for IoT services like pre-processing and prioritizing patients' data, running complex physiological models to analyze the processed information under context, which may have different QoS requirements and unpredictable resource consumption.

There are many works in the literature for general resource allocation in cloud computing environment as compared with the body area network domain. A green cloud framework has been proposed in [25]. The algorithm presented energy-efficient scheduling, virtual machine systems (VMS) image and image management components that are applied in four different ways. The authors emphasize the improvement of cloud infrastructure rather than applying different techniques with underlying hardware and software. The problem of resource pooling and allocation is addressed with the idea of geographically distributed datacenters connected using wireless networks [26]. The datacenters work in coordination and provide required resources to the requesting clients. Although this approach is cost-effective in terms of efficient resource utilization, however, the low bandwidth and high latency internet links are the bottleneck for the implementation. Resource and job scheduling algorithm based on Berge model has been proposed in [27]. It is based on two constraints. One is to classify user tasks by QoS preferences, and the second constraint is to define resource fairness justice function to judge the fairness of the resources' allocation. The resource allocation problem is attempted to be solved using a game theory model in [28]. This model considers criteria such as rational exchange behavior of cloud users, dynamic successive allocation, heterogeneous distribution of resources and incomplete common information.

Dynamic resource management in cloud using the migration of virtual machines has been proposed in [29]. It discusses the conditions when to migrate to a VM form one physical machine to another physical machine, which includes periodic maintenance, excess spare capacity, load imbalance and addition/removal of virtual machines and physical machines. It argues that resource constrained VMs should be migrated to a physical machine that has the capacity to host it. However, this paper does not provide any algorithmic solution nor an analytical model to solve the VM allocation problem. Recently, similar work has been proposed regarding resource allocation in CABAN environment. In [30], the authors proposed a fair energy management for IoT applications in cloud environment, which is based on VM migration technology. It aims to achieve the goal of energy savings and fairness improvement. The authors analyzed the energy consumption and the resource fairness in the cloud environment to formalize the problem. And a corresponding VM scheduling method was designed to solve this problem. The experimental results demonstrate the validity of their method.

There are some recent works which uses meta-heuristic algorithm such as genetic algorithm and particle swarm optimization for resource allocation in cloud and IoT. For example, a resource provisioning and scheduling strategy for scientific workflows on Infrastructure as a Service (IaaS) clouds has been discussed in [31]. Particle swarm optimization (PSO) and meta-heuristics optimization techniques has been used in this algorithm. A new resource allocation and scheduling technique based on genetic algorithm has been proposed in [32]. Before deploying any resource calculated by the algorithm, it computes the influence that the resources will have on the system and select the least effective one. In support of the algorithm, an analytical model is also provided. However, this paper does not provide a proof for the time a genetic algorithm takes to compute different deployment strategies.

In [33] and [34], the authors proposed an approach to allocate a bag of tasks using genetic algorithm. Considering the limitations of budget and deadlines, the genetic algorithm performed effectively. However, it is found that the resource allocation based on genetic algorithm is quite slow in terms of time. The reason is the iterations involved in the algorithm to find the best possible solution. Similarly, genetic algorithm is used in [34] which is an approach to allocate cloud resources for IoT devices. IoT devices are used as medical equipment, which requires efficient processing from cloud. While the allocation is smooth and reliability of the process is very high, yet it suffers from slowness of genetic algorithm, which may be unbearable in life critical situations.

However, the above-mentioned approaches of resource allocation are inefficient to use in CABAN platform due to the complexity of this system that requires appropriate methods/techniques and technology. Such systems, in fact, consist of many distributed and interacting components that are usually heterogeneous in terms of hardware, communication protocols, software interfaces, and output data. In addition, the workloads in this platform have different priorities, performance and resource requirements. Therefore, there is a need of new approaches or techniques that can handle such complex systems.

B. AGENT BASED MODELLING FOR RESOURCE ALLOCATION IN CABAN HEALTHCARE PLATFORM

Now-a-days, agent-based modeling is also emerging as an effective solution for resource allocation in the domain of cloud computing and IoT due to its capability of handling such complex systems. However, some of the works focus on using agents in cloud resource allocation, while others focus on using agents for resource allocation in IoT platform. None of the works address to use ABM in CABAN platform with a healthcare scenario where it is more complex and challenging. For example, in [35], a multi-agent system is used to model the process of allocating of cloud resources. The authors proposed an approach based on agents to connect the different heterogeneous cloud providers with the cloud users. The objective of the system is to allow users to choose the appropriate resources that can meet their needs. Several experiments show that autonomous agents make the clouds smarter in their interactions with users and more efficient in resources allocation. The only metric used for the validation phase is the cost of services or applications, while studying the performance of a system is usually measured by the speed of sending the response to a user.

A similar observation of efficiency of agent-based approach is reported in [36]. The study considered the allocation of virtual machines to the requesting clients to physical datacenters. However, the simulations are done on CloudSim platform, which is not an ABM tool for modeling. The results provided in [36] could be improved by using ABM tool such as Netlogo or Starlogo to model the resource VM allocation process in the cloud. In [37], the authors proposed an adaptive resource allocation model by implementing an agent-based test bed that allocates the consumer's job to an appropriate data center based on network delay between a consumer and data center and the workload of each data center in cloud computing environment. The results show a better response time for allocation as compared to some existing resource allocation models.

In our work, we also utilize ABM to allocate resources in CABAN platform. However, our work is enhanced in a way that we consider greater details of resources such as memory and disk-spaces instead of only allocating virtual machine, which is important, especially in Healthcare scenario where it is hard to manage effectively not only the IoT devices but also resource information and process complicated queries with diverse factors. In particular, we proposed to use ontology with ABM to address above issues since an ontology uses a formal, logical knowledge representation that supports automated reasoning. There are very few works, which consider ontology-based resource allocation in the cloud [38]. However, none of the works consider using ABM and ontology for resource allocation in CABAN platform in pervasive healthcare scenario, which has diverse QoS parameters that globally affect the overall cloud resource's utilization. In our work, we also categories two modes of resource allocation which are general and emergency modes. Detailed results related to reliability of the resource allocation process, efficient utilization of the resource and robustness of the process are provided.

III. PROPOSED RESOURCE ALLOCATION MODEL

The CABAN platform employs various sensors to collect comprehensive physiological information and uses gateways and the cloud to analyze and store the information and then send the analyzed data wirelessly to caregivers for further analysis and review. Fig. 1 depicts the abstract system architecture of the proposed CABAN platform. The system is comprised of five main components: wearable sensors, mobile device, cloud servers, users and display terminals such as Television (TV), personal computer or smart phones.

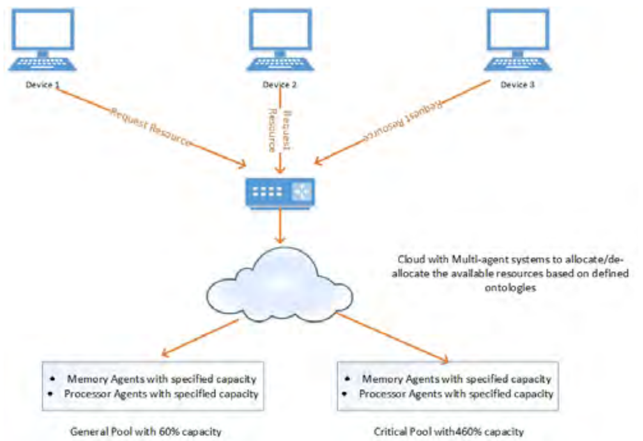


FIGURE 1. A general overview of the proposed resource allocation architecture in CABAN platform.

A. SYSTEM ARCHITECTURE OF THE PROPOSED RESOURCE ALLOCATION MODEL

Fig. 1 shows the logical architecture of the proposed resource allocation model. It is divided into three layers:

1-Client Layer: It consists of all medical devices which have data related to medical procedures. These data need to be processed for the information extraction. The client machines sent its requirements to the cloud in the form of memory and processing capacity required.

2-Network Layer: The intermediate network through which resource acquisition request is passed to the cloud. The higher the latency in the network, the more is the delay in response from the cloud.

3-Cloud Layer: It contains the memory and processing resources. It possesses the information of the status of each resource plus a mechanism to allocate resource when required. It has tools for handling pools of resources. One operates for general request while other operates for life-critical resource demands.

B. RESOURCE ALLOCATION ONTOLOGY

Fig. 2 shows the resource allocation ontology used by the proposed system. It defines agents (as requester and providers), Activities, Resources and relationship between them to describe how the usage of resources is coordinated. The coordination process presents many similarities with the resource allocation. For example, CPU and Memory components can be mapped to CPU and Memory agents.

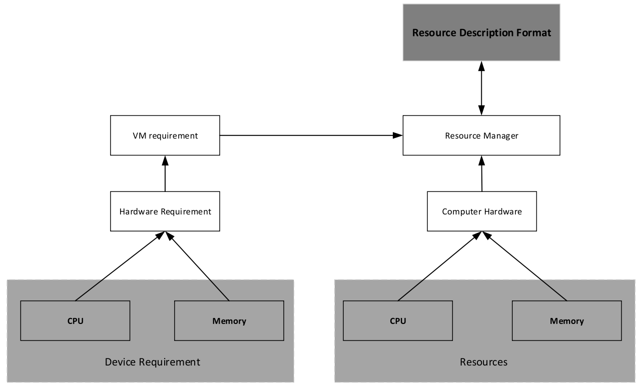


FIGURE 2. Resource Allocation Ontology.

The resource allocation process in a cloud based on ontologies is presented in [48]. It defines Requesters or providers as agents, activities, available resources and relations between them in order to describe the procedure for allocation and coordination of resources among them. The ontologies are described in Resource Description Format (RDF). The coordination process presents many similarities with the resource allocation. That is the reason that the classes such as Client Manager (CM) and Resource Manager (RM) can be described as agents and different client activities can be mapped as atomic activities. Only a set of resource and properties are defined to validate our semantic resource allocation. Only hardware resources such as CPU, memory, disk and network and software resources such as files are considered for different tasks.

Although the presented ontology partially describes the resource allocation system, yet it is incomplete. Therefore, some changes and extensions are required to make the coordination ontology suitable for resource allocation.

C. RESOURCE ALLOCATION BASED ON AGENTS

The proposed solution is divided into two major components, i.e. client side and server side. The client side have devices, request synchronizer and request forwarder. Server side has a process engine, rules database, log file and resource (processor and memory) allocation algorithm. An overview of the proposed model is shown in Fig. 3. The description of subcomponents is as follows:

1) MEDICAL DEVICES

Medical devices that need computing resources at runtime to process the complex medical procedures. These devices are normally used for Pathology, Chemical examination, Biochemistry and Cardiac. Traditionally, the data that needs to be processed manually is given to the standalone computing machines. Although this work but involve a life-critical delay.

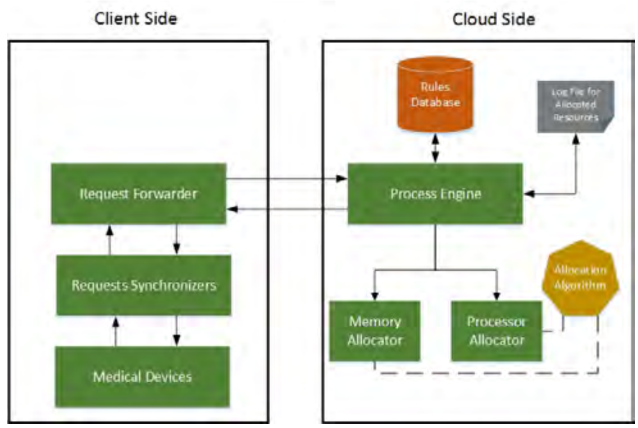


FIGURE 3. **Resource Allocation Model for Healthcare.**

2) REQUEST SYNCHRONIZERS

For a medium to a large-scale hospital, the demand for computing resources is significantly high during peak hours. This requires a mechanism to synchronize the requests based on the time stamp. Request synchronizer performs this action.

3) REQUEST FORWARDER

The request forwarder forwards the request. Any data structure such as First in First out (FIFO), Last in First out (LIFO) or Least Recently User (LRU) to forward the request in this module. FIFO is employed for the request forwarding.

4) PROCESS ENGINE

The main component at cloud side which handles the incoming resource allocation resources and allocates resources accordingly. The process engine makes decisions by extracting rules from the Rules database. The status of allocated resources is determined from the log file. The process engine requests resources from processor and memory agents.

5) RULES DATABASE

The rule's database contains rules in IF-THEN-ELSE structure to assist process engine for decision making. By following the modular approach, the Rule database can be updated with the new rules anytime without affecting rest of the system and the resource allocation process changes its behavior due to updated rules. The rules can be stored either in database or in a text file such as XML file. This database or file is thus an important part of an overall system. Agents dynamically evaluate the state of machines using these rules specified for the file allowing them to take decisions.

The rule file could be on client side (agent-side) or at the server side. For the client side, the rule file is provided to all clients, and the clients make decisions using the given rule file. However, the problem is updated in rule file, which requires unnecessary shutdowns to the systems. Second approach is server based where the rule file is kept on server and client access the server rule file to make any decision.

6) LOG FILE

The log file contains all the allocated resources with allocation starting time, allocation ending time and status (active, passive) of the resource. The process engine determines the status of resources from the log file and then decided to run the resource allocation procedure.

The log file stores the information in the form of sets for the condition of a resource (memory or processor) as follows:

$R = \{\text{occupied, free, healthy, ill}\}$

$T = \{\text{ts, te}\}$

Where,

t_s = Start time stamp

t_e = End time stamp

Start time stamp is the time when the resource is allocated and end time stamp is the time when the resource will be released.

7) MEMORY RESOURCES

Memory resources are memory agents that have attributes such as occupied, free, healthy, infected and capacity. The process engine continuously allocates and de-allocates the memory units based on their health and capacity.

8) PROCESSOR RESOURCES

Processor resources are processor agents that have attributes such as occupied, free, healthy, infected and capacity. The process engine continuously allocates and de-allocates the processing units based on their health and capacity.

D. RESOURCE ALLOCATION ALGORITHM FOR IoT HEALTHCARE DEVICES IN CLOUD PLATFORM

The proposed resource allocation algorithm for CABAN is shown in Fig. 4. The process is initiated by medical devices with certain requirements for data to process. This requirement is sent to requirement synchronizer module to resolve conflict between many requirements coming from different devices and then passed to the forwarder module which passes the requirement to the process engine. The process engine applies the rules from the rule's database and chooses which pool of VM resources (critical or general) is required. Then, an optimization algorithm is used to find the suitable servers for VM resource allocation. If the requested resources are not available to the pool, the process waits until resources get free. As soon as the resource allocated, an entry about the resources in use is made to the log file and the resource allocation procedure ends.

1) UNIQUE FEATURES OF PROPOSED RESOURCE ALLOCATION ALGORITHM

ABM based solutions have proven strengths in the domain of complex adaptive systems such as university examination scheduling and network's security. The developed resource allocation algorithm differentiates it from other existing solutions in following perspectives:

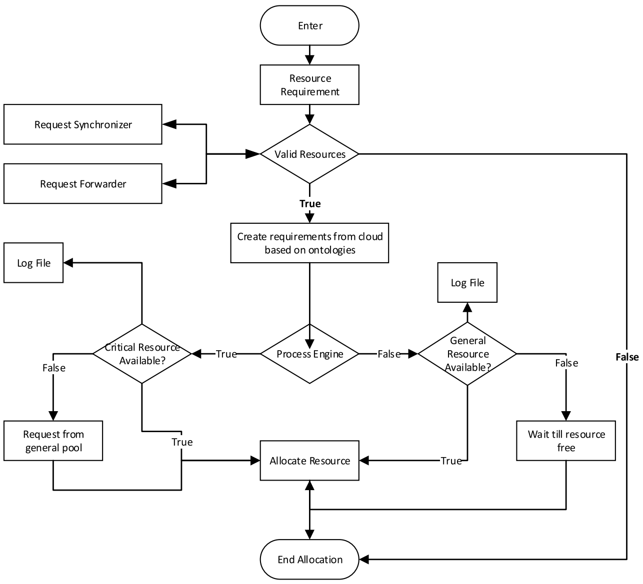


FIGURE 4. The Resource Allocation Algorithm.

1- Response Time: the resource time observed by the requesting devices should be less than the exiting algorithms. The reason is efficient management and serving of resources at management layer.

2- Maximum Resource Utilization: the resource allocation algorithm efficiently allocates/de-allocates the resources on cloud. This not only improves the utilization of resources in per unit time but also reduce the response time by requesting devices.

3- Robustness: the algorithm provides robust allocation and de-allocation of resources. This fact will be proved during experimentation phase when simulations will be executed for long time.

IV. PROPOSED OPTIMIZATION MODEL FOR RESOURCE ALLOCATION

The proposed VM resource allocation is modeled as an MILP optimization problem [39]. The objective is to discover the minimum number of physical servers to place all the VMs, as for physical server's abilities. We, likewise, consider two major requirements or constraints: improve resource use or utilization, and maintain the balancing of resource usage to avoid resource over-burden.

The parameters of linear programming formulation are presented in Table 1. For the VM resource requirements of medical devices I that need to be allocated, the MILP model is presented in Eq. (1) to (6).

$$\text{Minimize} \sum_{p\in P} y_p \tag{1}$$

$$\text{Subject to} \sum_{p\in P} x_{pv} = 1 \quad \forall v \in V \tag{2}$$

$$\sum_{v\in V} u_{vr} x_{pv} \le c_{pr} y_p \quad \forall p \in P, \forall r \in R \tag{3}$$

TABLE 1. The VM respirce allocation parameter.

Parameters	Description
P	set of physical servers
V	set of virtual machines
R	set of resources (CPU, memory)
$I = \{v_1, v_2, \ldots, v_n\}$	VM resource requirements of medical devices
$u_{vr},\ v \in R \,\&\, r \in R$	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 resource utilization on physical server $p \in P$ after allocating virtual machine $v \in V$
$d_{p_1p_2}$	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
$y_p \in \{0,1\}$	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

$$\sum_{v\in V} ruc_{pv} x_{pv} \le T_1 \quad \forall p \in P \tag{4}$$

$$\sum_{v\in V} s_{pv} x_{pv} \le T_2 \quad \forall p \in P \tag{5}$$

The target work in (1) goes for limiting the quantity of required physical servers. The limitation in (2) ensures that each virtual machine is mapped to a single physical server. Condition (3) ensures that the virtual machine requests allotted in each physical server doesn't over-burden its ability. Condition (4) can enhance the overall resource utilization. The constraint (5) can lessen the possibility of a resource over-burden and can conceivably adjust the balance of resource usage among every single physical server.

Since the future workload may not be unsurprising, the objective function in (1) depicts the normal insights from time 0 to current time. For the imperatives, they ought to be fulfilled whenever the designation choice is made. Therefore, we didn't present the time variable in those equations. The discretionary limitations (4) and (5) are proposed to diminish the hunting space down this NP-hard problem. In any case, the utilization of these obliges may lead the outcomes to be close ideal. The definitions and effectiveness of ruc_{pv}, S_{pv} and d_I are explained as follows:

Definition 1 (Resource utilization and balancing resource usage condition): Given a VM requirement of medical device i and a physical machine j, let c_{ij}, m_{ij} are the rates of asset utilization in regard to CPU and memory, separately. Thus, VM requirement of medical device i and virtual machine v can be utilized reciprocally. Let fc_j and fm_j be the rates of unused CPU and memory resources, on machine j. For any resource, if in any event $k\%$ of free limit is held to store the sudden workload burst, then it would not be considered in the accessible unused resources. The correct measure of the held resource is dictated by the cloud supplier by utilizing long haul benchmark or workload expectation models [40]. The VM resource of the medical device can be distributed to that physical machine just if the accompanying condition (6) is met:

$$c_{ij} \leq fc_j \& m_{ij} \leq fm_j \tag{6}$$

After the VM request of the medical device i is allocated on physical machine j, the average rate of unused or free resource ap_{ij} is calculated as follows (7):

$$ap_{ij} = (f\,c'_j + f\,m'_j)/2 \tag{7}$$

where,

$$f\,c'_j = fc_j - c_{ij}, f\,m'_j = fm_j - m_{ij} \tag{8}$$

The resource usage condition for machine j, after allocating VM request of the medical device i is denoted as ruc_{ij}, which is a mean-square value. Utilizing (9), we have characterized ruc_{ij} in (4) as follows

$$ruc_{ij} = (f\,c'_j - ap_{ij})^2 + (f\,m'_j - ap_{ij})^2 \tag{9}$$

We expect that the applications running on a solitary physical machine share the greater part of the resource capacity relatively. All the more particularly, medical device i will be distributed on the physical machine j, which will get $(c_{ij}/(1-f\,c'_j))\times 100\%$ of the CPU limit, and same for memory assets.

As the resources may not be dealt with similarly in healthcare framework, requirement (5) is made to address this issue. Give us a chance to consider a case where CPU is more imperative for handling the medical device tasks, for example, confront acknowledgment or information investigation. As to i, the prevalence of machine j is characterized as s_{ij} in (5) by utilizing (10):

$$s_{ij} = c_{ij}/(c_{ij}/(1 - f\,c'_j)) = 1 - f\,c'_j \tag{10}$$

A. DETERMINATION THE THRESHOLD VALUE

In the proposed optimization problem, there are two types of threshold: overall resource utilization condition T_1 in (4), and free resource conditions T_2 in (5). T_2 will be generated according to the specific QoS requirement, the benchmark or the workload burst prediction regarding each medical device VM request requirement. For T_1, we use the objective function presented in (1) to determine the optimal value. However, if the physical servers are not capable of supporting all of the requests, in order to find the optimal value of T_1, we will introduce a new objective function U using (11):

$$U = \lambda_c[\sum_{p_j\in P}(1-fc_j)\times oc_j] + \lambda_m[\sum_{p_j\in P}(1-fm_j)\times om_j] - C_{P'} \tag{11}$$

where U is the overall resource utilization; oc_j and om_j are the total amount of CPU, and memory on physical server p_j; λ_c and λ_m denote the pricing schemes for each CPU unit and memory unit respectively; $C_{P'}$ is the total cost for maintaining the active physical servers. This new objective function measures the total server resource utilization. For the simplicity of explanation, we describe the threshold selection algorithm with the objective function U. It is exactly same as if the objective function U is replaced by the one that we presented in (1).

V. EVALUATION

In this section, we present our evaluation methodology and simulation results of the proposed resource allocation algorithm in CABAN framework for Smart home healthcare scenario. We used NetLogo simulation tool for agent-based resource allocation model. It defines an environment for simulating natural phenomena by using four types of entities: turtles (mobile agents), patches (static agents), links and the observer. It has visual output, which helps to understand the overall process. Moreover, it has a rich environment of built-in functions and GUI controls that allow rapid development of agent-based models. Furthermore, the selected tool Netlogo is related to the Protege extension specifically designed to perform interoperability between OWL and Netlogo [41]–[43]. Thus, it is possible to export ontologies to Netlogo form OWL and vice-versa.

A. PERFORMANCE OF THE PROPOSED RESOURCE ALLOCATION ALGORITHM

In NetLogo, the identified agents of this resource allocation system following ontologies are as follows:

1-VM Agent: This agent handles the request of the medical device (also an agent) for required resources to process the data in the cloud.

2-Device Agent: It is client side or requestor agents that create a requirement for memory and processing. These requirements are sent to the cloud for allocation, and resources are allocated based on urgent or general requirement.

3-Memory Agent: It keeps track of available and in used memory for overall cloud. It continuously allocates memory to the requestor and de-allocates when its usage time limit expires.

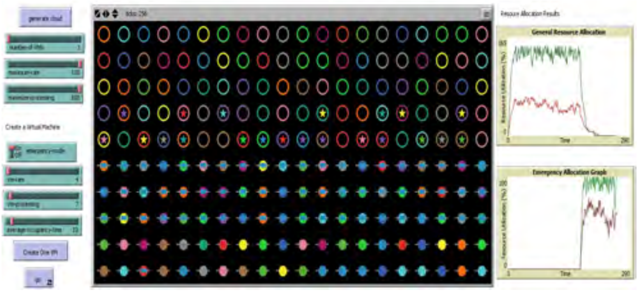


FIGURE 5. Resource allocation implementation basic GUI.

4-Processor Agent: It keeps track of processing resources of the cloud. The resources are allocated and de-allocated continuously on demand from the requesting client side devices.

The modes of operation of devices are emergency and general mode. The general mode contains 60% of processing and memory resources on cloud while an emergency mode contains the 40% of the resources on cloud. The graphics user interface (GUI) of the Netlogo with all controls is shown in Fig. 5.

The resources are divided into memory and processing resources. Memory resource represents one unit of memory (KB or MB or GB) and denoted by a colored circle. The processor resource represents one unit of processor (KHz, MHz or GHz) and denoted by blue line. First, three rows represent general resources while last two rows represent emergency resources. The algorithm operates in one mode at one time which is either general or emergency.

Initially, a seed value of random-access memory (RAM) and processing resource required by a virtual machine for a defined interval of time are set from the graphical user interface. The procedure is started, and VMs are continuously allocated and de-allocated based on the available resources for the corresponding resource pool on cloud. The results from the simulation are shown in Fig. 6.

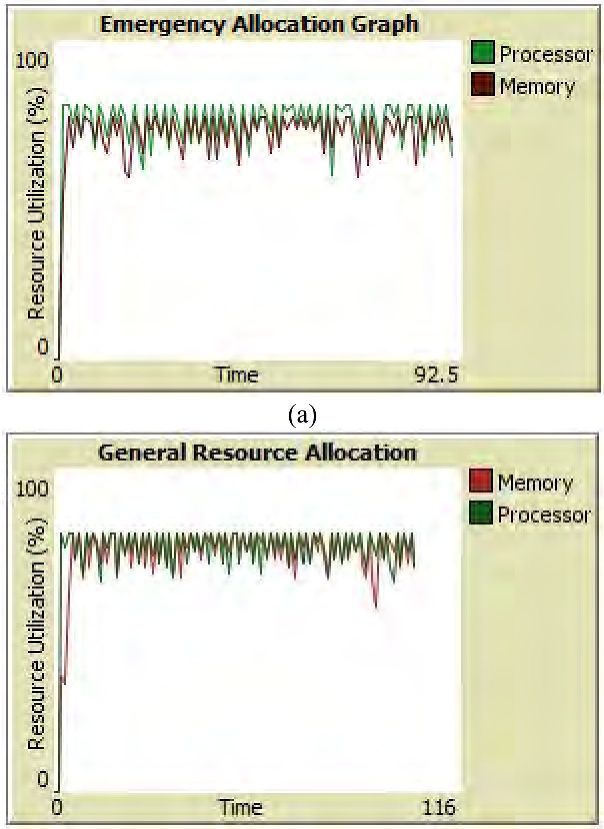
The resources that are allocated at each instant of time can be recorded in a file. These results can be exported to CSV, Excel or spreadsheet formats or in the form of table using the graphical user interface provided by Netlogo. The tool used by Netlogo to export results is called 'Behavior Space', and a sample of the export utility is shown in Figure 6. The application has been set up for result's generation with following constraints:

1- The resources in cloud are fixed by 100 units of memory resources and 100 units of processing resources.

2- The resources will be allocated randomly from the available resources in the pool. No choice it provided to select a particular resource for allocation. The probability of resource selection by the algorithm is equally likely.

3- The output is visual only. It is showing the resource allocation graph as well. However, no code is written to export the results to output file.

In Netlogo, all the resources such as memory, processors, number of VMs and time required by each VM are considered



(a)

(b)

FIGURE 6. The resource utilization graphs (a) emergency mode and (b) general mode.

as input. These resources could be managed from the GUI controls provided for the application. Two different modes of the resource allocation in a cloud are discussed, and the detail of results is provided in the subsequent sections:

1) Case 1: Constant Units of Memory and Processor

Constant units of memory and processing resources required by the VM are discussed throughout this section. The sample input is shown in the Table 2.

TABLE 2. SAMPLE INPUT FOR CASE 1.

Memory	Processor	Occupancy Times
1000	1000	1000 units

Visual output can be seen on the Netlogo GUI. The results can be exported either writing particular code for data exporting in output file or a specialized tool such as 'behavior space can be used to export the results in tabular and CSV file format. This tool requires separate configuration to export results in any available output formats. For the sake of achieving results of a particular problem with the thesis, only one execution is performed to allocate the required resources. However, the Behavior Space tool is used to run multiple

TABLE 3. Sample Input for case 2.

Memory	Processor	Occupancy Times
770	690	695

TABLE 4. Response time in milliseconds for cases 1 & 2.

No	Proposed Algorithm (m sec)		Genetic Algorithm (m sec)	
	Case 1	Case 2	Case 1	Case 2
1	0.141	0.156	0.234	0.344
2	0.047	0.078	0.156	0.141
3	0.078	0.063	0.141	0.109
4	0.078	0.078	0.141	0.094
5	0.078	0.078	0.157	0.094
6	0.063	0.093	0.109	0.11
7	0.062	0.093	0.109	0.078
8	0.062	0.094	0.125	0.093
9	0.062	0.078	0.109	0.109
10	0.063	0.062	0.14	0.078

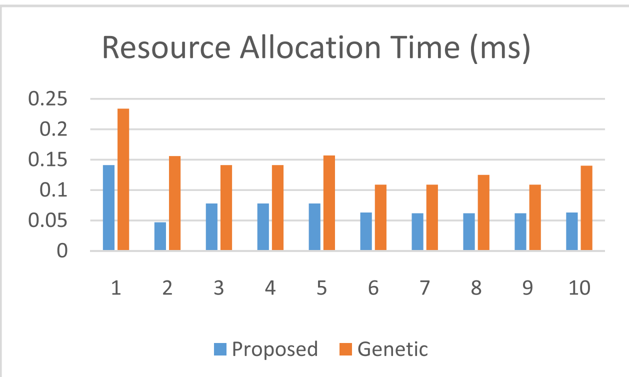


FIGURE 7. Resource Allocation Time for Case 1.

executions of the same code to predict the actual behavior during the program.

2) Case 2: Random Units of Memory and Processor

The resource allocation is performed for random number of resources with number of units less than 1000. The details of resources are shown in Table 3.

The results obtained for different cases will be evaluated for the metrics such as response time, robustness and maximum resource utilization. The results are comparatively evaluated with the Genetic algorithm which is based on random population. In genetic algorithm, the selection of a resource

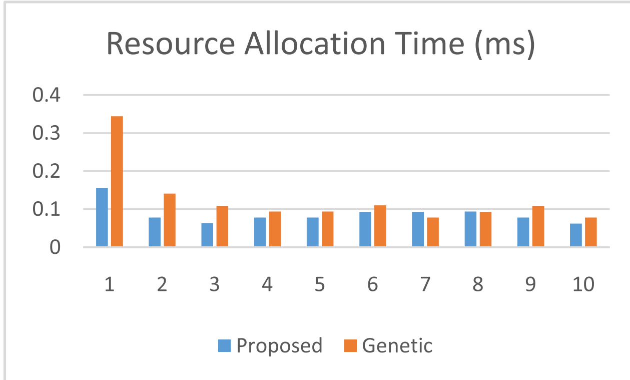
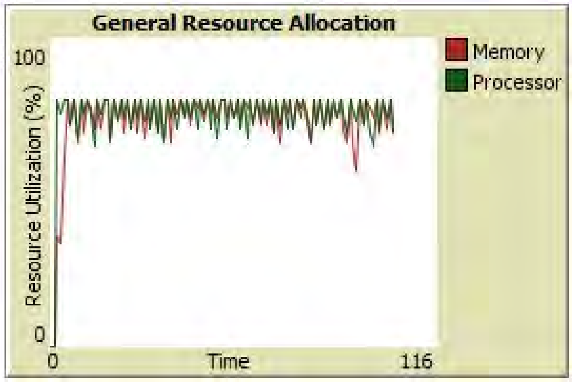
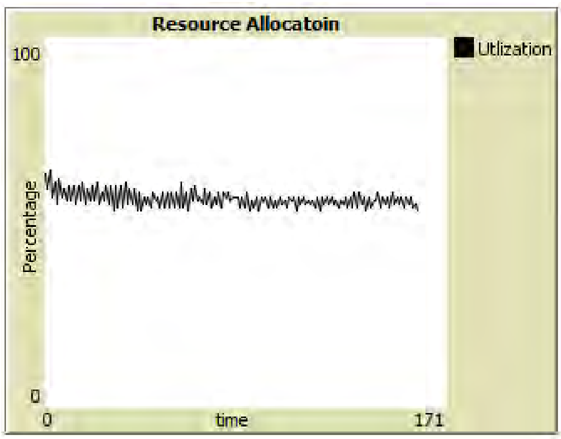


FIGURE 8. Resource Allocation Time for Case 2.



(a)

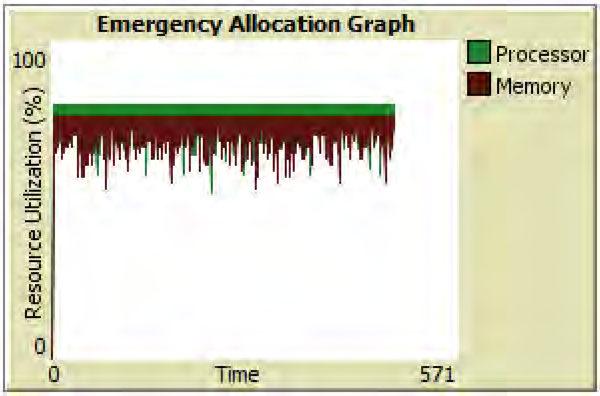


(b)

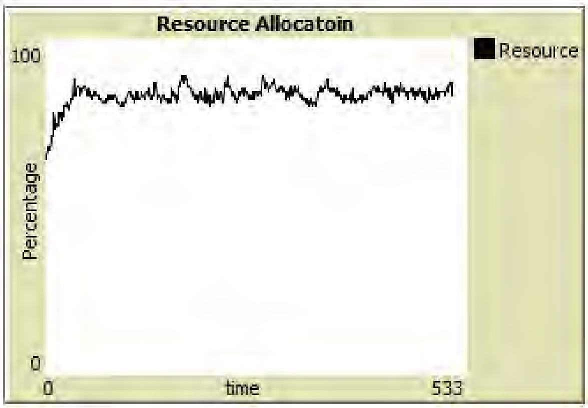
FIGURE 9. (a) General Resource Allocation. (b) GA Resource Allocation.

is based on a fitness function. Multiple mutations are performed to select the best resource. The dynamic replacement of resources and time frame has not been covered in this study. Both algorithms are implemented using Netlogo.The application executed 10 times each for case 1 and case 2.

The execution time of both of the cases for each of the run is shown in Table 4. A comparison analysis for 10 executions of each case (case 1 and case 2) is shown in Fig. 7 and Fig. 8.



(a)



(b)

FIGURE 10. **(a) ABM Robustness. (b) GA Robustness.**

Response time by the resource allocation algorithm to serve one request is less than one millisecond for both cases in both algorithm. The average execution time for case 1 is 0.0734 and 0.1421 milliseconds for the proposed algorithm and genetic algorithm respectively. Similarly, the average execution for case 2 is 0.0873 and 0.125 milliseconds for proposed algorithm and genetic algorithm respectively.

It is observed that, for constant number of resources and random number, the performance of proposed algorithm is comparatively better than the genetic algorithm. This is due to the number of mutations performed in genetic algorithm to produce results. The results proved the effectiveness of Netlogo to model resource allocation being a complex problems. It is to be noted that the slight increase in time for case 2 as compared with case 1 in the proposed algorithm is due to the computation of random values for the resources.

The Netlogo results demonstrate very promising resource utilization results. Almost all the resources are efficiently utilized for both modes i.e. proposed and genetic. The graphs showing the percentage of resource utilization are shown in Fig. 9.

The graphs show the maximum utilization of available resources with efficient allocation and de-allocation for significantly large time interval (184 units for general allocation in proposed algorithm). However, the resource allocation process stops after allocating resources up to 170 units of time. This is due to inefficient memory handling and multiple iterations of genetic algorithm that stops the simulation after some time. However, using the proposed algorithm, no simulation stoppage/crash observed.

The continuous resource allocation and de-allocation may cause bottleneck created by algorithm and cause improper termination of the process. The under-discussion algorithm is test for robustness by making the resource utilization to maximum and observe the process for around 500 units of time. This process is repeated for 20 times and no improper termination of algorithm is observed. Fig. 10 shows the result of the experiment. This proves the robustness of algorithm under peak-load values. Genetic algorithm is found to be as much robust as the ABM algorithm. However, GA algorithm is not efficient in terms of time as compared with ABM algorithm due to iterations that it performs during allocation process.

VI. CONCLUSION

This paper mainly tackles the research challenges on how to efficiently manage and allocate cloud resources in accordance with the QoS requirement of pervasive healthcare services and applications in CABAN platform. QoS is a challenging requirement in CABAN healthcare scenario where the delay in treatment for critical patient can create a difference of life and death. To address the problem of cloud resource allocation in this complex framework, a fast and robust resource allocation algorithm is proposed by using Agent Based Modeling (ABM) and ontology. We combine ontology with ABM and show the relation between them. In order to evaluate the efficiency of the proposed solution, an analysis is also performed based on provided input dataset. The evaluation is performed by measuring the execution time of the algorithm that allocates the resources based on given input dataset. A comparative analysis of the proposed solution using agent-based modeling and other existing resource allocation tools, techniques and algorithm based on different theories has been provided. The results demonstrate that the complex problems can be solved using agent based modeling (ABM). The usability and practicality of ABMs has been proved by implementing the solution. In future, we will test the algorithm by considering more resources such as GPU and bandwidth in a real-world scenario.

REFERENCES

[1] L. Liu, E. Stroulia, I. Nikolaidis, A. Miguel-Cruz, and A. R. Rincon, "Smart homes and home health monitoring technologies for older adults: A systematic review," *Int. J. Med. Inf.*, vol. 91, pp. 44–59, Jul. 2016.

[2] M. Memon, S. R. Wagner, C. F. Pedersen, F. H. A. Beevi, and F. O. Hansen "Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes," *Sensors*, vol. 14, no. 3, pp. 4312–4341, 2014.

[3] N. M. Lopez, S. Ponce, D. Piccinini, E. Perez, and M. Roberti, "From hospital to home care: Creating a domotic environment for elderly and disabled people," *IEEE Pulse*, vol. 7, no. 3, pp. 38–41, May 2016.

[4] J. Han, C.-S. Choi, W. K. Park, and I. Lee, and S. H. Kim, "Smart home energy management system including renewable energy based on ZigBee and PLC," in *Proc. IEEE Int. Conf. Consumer Electron. (ICCE)*, Jan. 2014, pp. 544–545.
[5] F. Cicirelli, G. Fortino, A. Giordano, A. Guerrieri, G. Spezzano, and A. Vinci, "On the design of smart homes: A framework for activity recognition in home environment," *J. Med. Syst.*, vol. 40, no. 9, pp. 200-1–200-17, 2016.
[6] Z. J. Kuang, L. Hu, and F. Y. Chen "Analysis Markov delay control strategy for smart home systems," *Appl. Mech. Mater.*, vol. 484, pp. 413–417, Jan. 2014.
[7] H. Ghayvat, S. C. Mukhopadhyay, and X. Gui "Sensing technologies for intelligent environments: A review," in *Intelligent Environmental Sensing*. Cham, Switzerland: Springer, 2015, pp. 1–31.
[8] I. Azimi, A. M. Rahmani, P. Liljeberg, and H. Tenhunen, "Internet of things for remote elderly monitoring: A study from user-centered perspective," *J. Ambient Intell. Humanized Comput.*, vol. 8, no. 2, pp. 273–289, Apr. 2017.
[9] M. M. Hassan, K. Lin, X. Yue, and J. Wan, "A multimedia healthcare data sharing approach through cloud-based body area network," *Future Generat. Comput. Syst.*, vol. 66, pp. 48–58, Jan. 2017.
[10] G. W. Lee, S. H. Na, K. H. Kim, and E. N. Huh, "Cloud-based smart home system (CbSH) architecture design for virtual home gateway and cloud interworking," in *Proc. 3rd Int. Conf. Convergence Technol.*, Jul. 2013, pp. 1572–1573.
[11] X. Ye and J. Huang, "A framework for cloud-based smart home," in *Proc. Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Dec. 2011, pp. 894–897.
[12] Y. Lyu *et al.*, "High-performance scheduling model for multi-sensor gateway of cloud sensor system-based smart-living," *Inf. Fusion*, vol. 21, pp. 42–56, Jan. 2015.
[13] R. Gravina and G. Fortino, "Automatic methods for the detection of accelerative cardiac defense response," *IEEE Trans. Affect. Comput.*, vol. 7, no. 3, pp. 286–298, Sep. 2016.
[14] G. Fortino, S. Galzarano, R. Gravina, and W. Li, "A framework for collaborative computing and multi-sensor data fusion in body sensor networks," *Inf. Fusion*, vol. 22, pp. 50–70, Mar. 2015.
[15] G. Fortino, D. Parisi, V. Pirrone, and G. di Fatta, "BodyCloud: A SaaS approach for community body sensor networks," *Future Generat. Comput. Syst.*, vol. 35, pp. 62–79, Jun. 2014.
[16] G. Fortino, G. di Fatta, M. Pathan, and A. V. Vasilakos, "Cloud-assisted body area networks: State-of-the-art and future challenges," *Wireless Netw.* vol. 20, no. 7, pp. 1925–1938, 2014.
[17] I. Sameer, "A framework for creating healthcare monitoring applications using wireless body sensor networks," in *Proc. BODYNETS*, 2008, p. 8.
[18] A. Andreoli, R. Gravina, R. Giannantonio, P. Pierleoni, and G. Fortino, "SPINE-HRV: A BSN-based toolkit for heart rate variability analysis in the time-domain," in *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment*. Berlin, Germany: Springer, 2010, pp. 369–389.
[19] F. Amblard *et al.*, "Introduction to NetLogo," *Agent-Based Spatial Simul. Netlogo*, vol. 1, pp. 75–123, Nov. 2015.
[20] M. Hoffmann, "Agent-based modeling," *Qualitative Methods in International Relations. A Pluralist Guide*. Berlin, Germany: Springer, 2008, pp. 187–210.
[21] R. Conte *et al.*, "Manifesto of computational social science," *Eur. Phys. J., Special Topics*, vol. 214, no. 1, pp. 325–346, 2012.
[22] E. Bevacqua, M. Mancini, and C. Pelachaud, "A listening agent exhibiting variable behavior," in *Intelligent Virtual Agents*, vol. 5208. Berlin, Germany: Springer, 2008, pp. 262–269.
[23] M. Swan, "Sensor mania! The Internet of Things, wearable computing, objective metrics, and the quantified self 2.0," *J. Sens. Actuator Netw.*, vol. 1, no. 3, pp. 217–253, 2012.
[24] G. M. Lee and N. Crespi, "Resource provisioning using local and public clouds for Internet of Things," in *Proc. 6th Meeting KuVS Fachgespräch NGSDP*, Berlin, Germany, Apr. 2012.
[25] A. J. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," in *Proc. Int. Conf. Green Comput.*, Aug. 2010, pp. 357–364.
[26] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe, "CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines," *ACM Sigplan Notices*, vol. 46, no. 7, pp. 121–132, Mar. 2011.
[27] B. Xu, C. Zhao, E. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Adv. Eng. Softw.*, vol. 42, no. 7, pp. 419–425, Jul. 2011.
[28] F. Teng and F. Magoulès, "A new game theoretical resource allocation algorithm for cloud computing," in *Advances in Grid and Pervasive Computing (GPC)* (Lecture Notes in Computer Science), vol. 6104. Berlin, Germany: Springer, 2010.
[29] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Commun. Mag.*, vol. 50, no. 9, pp. 34–40, Sep. 2012.
[30] G. Xing, X. Xu, H. Xiang, S. Xue, S. Ji, and J. Yang, "Fair energy-efficient virtual machine scheduling for Internet of Things applications in cloud environment," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 2, pp. 1–11, 2017.
[31] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 222–235, Apr./Jun. 2014.
[32] J. Gu, J. Hu, T. Zhao, and G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment," *J. Comput.*, vol. 7, no. 1, pp. 42–52, Jan. 2012.
[33] J. O. Gutierrez-Garcia and K. M. Sim, "GA-based cloud resource estimation for agent-based execution of bag-of-tasks applications," *Inf. Syst. Frontiers*, vol. 14, no. 4, pp. 925–951, 2012.
[34] M. Kim and I.-Y. Ko, "An efficient resource allocation approach based on a genetic algorithm for composite services in IoT environments," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2015, pp. 543–550.
[35] M. E.-K. Fareh, O. Kazar, M. Femmam, and S. Bourekkache, "An agent-based approach for resource allocation in the cloud computing environment," in *Proc. 9th Int. Conf. Telecommun. Syst. Services Appl. (TSSA)*, Nov. 2015, pp. 1–5.
[36] A. Al-Ou'n, M. Kiran, and D. D. Kouvatsos, "Using agent-based VM placement policy," in *Proc. 3rd Int. Conf. Future Internet Things Cloud, Int. Conf. Open Big Data (OBD)*, Aug. 2015, pp. 272–281.
[37] G. Jung and K. M. Sim, "Agent-based adaptive resource allocation on the cloud computing environment," in *Proc. 40th Int. Conf. Parallel Process. Workshops*, Sep. 2011, pp. 345–351.
[38] Y. B. Ma, S. H. Jang, and J. S. Lee, "Ontology-based resource management for cloud computing," in *Proc. Asian Conf. Intell. Inf. Database Syst.*, Apr. 2011, pp. 343–352.
[39] M. M. Hassan *et al.*, "Efficient virtual machine resource management for media cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 5, pp. 1567–1587, May 2014.
[40] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013.
[41] P. Livet, J.-P. Müller, D. Phan, and L. Sanders, "Ontology, a mediator for agent-based modeling in social science," *J. Artif. Soc. Social Simul.*, vol. 13, no. 1, pp. 1–12, 2010.
[42] J. G. Polhill, "Extracting owl ontologies from agent-based models: A netlogo extension," *J. Artif. Soc. Social Simul.*, vol. 18, no. 2, p. 15, 2015.
[43] J. Ejarque *et al.*, "Using semantics for resource allocation in computing service providers," in *Proc. IEEE Int. Conf. Services Comput.*, Jul. 2008, pp. 583–587.

MOHAMMAD MEHEDI HASSAN (M'12) received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2011. He is currently an Associate Professor with the Information Systems Department, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia. He has authored or co-authored over 100+ research papers in the journals and conferences of international repute. His research areas of interest are cloud federation, multimedia cloud, sensor-cloud, Internet of Things, Big data, mobile cloud, cloud security, IPTV, sensor network, 5G network, social network, publish/subscribe system, and recommender system. He has served as the Chair and a Technical Program Committee Member in numerous international conferences/workshops, such as the IEEE HPCC, ACM BodyNets, the IEEE ICME, the IEEE ScalCom, ACM Multimedia, ICA3PP, the IEEE ICC, TPMC, and IDCS. He received the Best Paper Award from the CloudComp Conference at China in 2014. He also received the Excellence in Research Award from CCIS, KSU in 2015 and 2016, respectively. He has also played role of the Guest Editor of several international ISI-indexed journals.

HANOUF ALBAKR is currently pursuing the M.S. degree with the Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. Her research interests include smart home, pervasive healthcare, Internet of Things, and body area network.

HMOOD AL-DOSSARI is currently an Assistant Professor with the Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His current research interests are data mining and knowledge discovery, quality of service assessment, reputation and trust management, sentiment analysis and opinion mining, big data analytics, and human–computer interaction.

AMR MOHAMED (SM'14) received the M.S. and Ph.D. degrees in electrical and computer engineering from The University of British Columbia, Vancouver, BC, Canada, in 2001 and 2006, respectively. He has over 20 years of experience in wireless networking research and industrial systems development. He is currently an Associate Professor with the College of Engineering, Qatar University, and the Director of the Cisco Regional Academy. He has authored or co-authored over 120 refereed journal and conference papers, textbook, and book chapters in reputed international journals, and conferences. His research interests include networking and MAC layer techniques mainly in wireless networks. He has served on the organization committee of many other international conferences as a TPC Member, including the IEEE ICC, GLOBECOM, WCNC, LCN, and PIMRC, and a technical reviewer for many international publishers, such as IEEE, ACM, Elsevier, Springer, and Wiley journals. He received three awards from IBM Canada for his achievements and leadership, and three best paper awards, latest from the IEEE/IFIP International Conference on New Technologies, Mobility, and Security 2015 in Paris. He has served as a Technical Program Committee (TPC) Co-Chair for workshops in the IEEE WCNC'16. He has served as a co-chair for technical symposia of international conferences, including Globecom'16, Crowncom'15, AICCSA'14, the IEEE WLN'11, and the IEEE ICT'10.