Received: November 17, 2016

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Towards Maximum Resource Utilization and Optimal Task Execution for Gaming IoT Workflow in Mobile Cloud

Shakkeera L[1]* **Latha Tamilselvan[2]**

[1,2]B.S. Abdur Rahman University, Chennai-600048, Tamilnadu, India
* Corresponding author's Email: lshakeera@bsauniv.ac.in

Abstract: With the rapid and tremendous growth of real-time resource discovery process ubiquitously, Internet of Things (IoT) and mobile cloud technologies have become an essential part of the future internet. The adaptive and energy-efficient mobile application execution is still in its infancy stage due to the diversity of IoT applications, manipulating workflow of an application and handling a large volume of streaming data in the heterogeneous resource-rich environment. To overcome these constraints, this paper introduces a GAming IoT workflow Fuzzifiers and Supporting MAximum Resource utilization and optimal Task execution on mobile cloud (GAF-SMART) paradigm offering an intelligent way of computing the mobile applications on the remote server. The GAF-SMART employs Game-theory and Fuzzy logic assisted adaptive task scheduler to attain optimal execution for maintaining a workflow of an application. Thus, the proposed mobile cloud IoT paradigm offers high Quality of Service (QoS), increases resource utilization and minimizes the mobile device energy.

Keywords: Mobile cloud, IoT, Game-theory, Fuzzy logic, Adaptive task scheduler, Workflow model, Non-dominant.

1. Introduction

Smartphones popularity has been drastically increasing over the past decade. The new era of mobile computing technology promises the enhanced computing capabilities of smartphones in terms of feature support for customized user applications, multimodal connectivity storage capacity and battery lifetime. However, mobile devices are still inherently limited by computing ability, low bandwidth and battery lifetime. To overcome the obstacles of ever-increasing computational and energy demands of sophisticated smartphone applications, the researchers envision the computing capability of computational clouds for mobile devices. Mobile Cloud Computing (MCC) [1, 2] leverages the cloud application processing services to mitigate the resource limitations of the smartphones, which provide a virtual computing environment to the mobile devices. In MCC, the application offloading method is considered as a meaningful solution for sharing the workload of the smartphones [3].

Recently, the rapid technological advancement has provided a feasible environment for connecting the variety of smart mobile devices together over the Internet to enable data interoperability methods for a specific real-time application. The IoT [4] is the dynamic and global network of interconnected intelligence and self-configuring mobile devices, enabling the ubiquitous computing scenario in the real-world. IoT is characterized by network embedded devices with limited processing and storage capacity and it encounters the consequential issues regarding responsiveness, performance and reliability. IoTs are recognizable, identifiable, addressable, controllable over the Internet in which IoT objects include smartphones, laptops, personal computer, household appliances, video systems, ambient devices, RFID tagged objects, wireless sensor networks and intelligent buildings.

The CloudIoT paradigm has been accepted in a variety of applications such as healthcare, smart city, smart home, video surveillance, automotive and smart mobility, smart energy and so on [5]. With the dramatic growth of resource-hungry IoT and multimedia mobile applications, MCC has become a significant research topic in scientific and industrial communities. The integration of IoT and mobile cloud [6] establishes greater impact in providing the services in a distributed and dynamic manner. Supporting the dynamic stream of data over the IoT devices is a challenging task in mobile cloud IoT environment. Hence, the mobile cloud IoT system necessitates the smart scheduling and allocation algorithms to intelligently perform the mobile cloud computations while conserving the device energy.

Our main contributions are summarized as follows:

- The main objective of the GAF-SMART is to obtain the adaptive cloud-based IoT mobile application execution and provide better QoS to the mobile cloud user while ensuring the negligible SLA violations.
- The GAF-SMART maps the application logic and determines the corresponding workflow model for the cloud tasks offloaded by the Cloudlet architecture to utilize IoT cloud information center.
- Exploiting the game-theory based fuzzy logic that optimally mitigates the execution time on the remote server and reduces the dominance value for each application.
- The hierarchical model based adaptive scheduling method tackles the dominance constraint without ignoring the workflow of an application.
- The experimental results reveal that the GAF-SMART accomplishes the energy-efficient IoT application execution while offering timely responses to the end-users.

1.1 Paper organization

The organization of the rest of the paper is as follows: Section 2 reviews the earlier works related to the task scheduling and resource allocation techniques and IoT application execution techniques in a mobile cloud environment. Section 3 describes the proposed GAF-SMART methodology. Section 4 discusses the evaluation results of the proposed system. Finally, Section 5 presents the conclusion and future work.

2. Related works

In MCC, the earlier works comprehensively present the different frameworks and architectures for effective code offloading and optimal execution of the mobile applications. A framework in [7] addresses the partitioning problem in mobile data stream applications and maximizes the performance of the MCC by exploiting the centralized heuristic genetic algorithm. Dynamic programming based offloading algorithm (DPOA) [8] optimally partitions the mobile applications to take the optimal offloading decision. Evidence-based Mobile Cloud Offloading (EMCO) approach [9] exploits the fuzzy sets to make the mobile code offloading decision by considering the processing capabilities of the mobile and the cloud. However, a lot of earlier efforts in computation offloading method, offering optimal execution in a mobile cloud environment is still at its beginning stage.

Energy-aware Dynamic Task Scheduling (EDTS) algorithm [10] utilizes the Dynamic Voltage Scaling (DVS) technique on the runtime environment to mitigate the energy consumption of the smartphones while satisfying the probability constraint and stringent time constraint of the mobile applications. Energy-aware Mobile Application Consolidation and Scheduling (E-MACS) algorithm [11] consolidates the application using load balancing techniques to improve the QoS and minimizes the overall energy consumption, cost of application migration and response latency. A randomized algorithm [12] determines the optimal solution for QoS-aware task allocation by selecting the appropriate cloud provider for each application execution. Game-theoretic resource allocation framework [13] allocates the resources based on the device energy. It employs the Nash equilibrium algorithm to resolve the energy minimization problem as the congestion game in a mobile cloud environment. However, these techniques lack in considering the response time and energy as the important metrics in dynamic task scheduling and resource allocation for streaming applications.

Several conventional research works on cloud-based IoT management schemes [14] investigate the outsourcing, task scheduling and resource allocation problem to tackle the shortcomings in mobile cloud IoT framework. An effective approach [15] exploits a learning technique and Markov decision process to dynamically assess the mobile IoT applications and enhance the efficiency of the IoT device respectively using the cloud computing technology based intelligent planning method. Nested Game based offloading method for MCIoT system (NG-MCIoT) [16] employs the Rubinstein game approach to determine the offloading portion of the computation and dynamically allocates the computing resources to the offloaded tasks on the remote server.

However, the existing mobile cloud techniques require the additional knowledge of handling the IoT mobile applications. Even though the current mobile cloud research presented different scheduling and allocation methods for IoT mobile applications, it is essential to appropriately identify the stream data for the corresponding request from the various stream data generated by the IoT devices. Also, analyzing the utilization of the cloud resources is significant while dynamically scheduling the mobile IoT applications. Moreover, most of the IoT mobile applications are time-sensitive applications, which demand a quick response from the remote server. Therefore, developing an adaptive system is necessary to handle the dynamic data changes and to ensure seamless latency-sensitive application execution in the mobile cloud IoT environment.

3. GAF-SMART methodology

The dynamic and rapid growth of the IoT applications frequently increases the amount of streaming information. Hence, the mobile device is limited to store and share the data itself due to its resource-scarcity. The mobile devices leverage to continuously interact with the surrounding IoT devices to perform the real-time resource discovery and the data retrieval process, which is a resource-intensive process for the mobile devices. To overcome this obstacle, the IoT mobile applications necessitate the cloud computing services which provide the offloading, storage and computing services to the mobile device. Instead of dynamically scheduling the tasks in the mobile cloud environment, the IoT applications require the adaptive data analytics and monitoring process to ensure the workflow dependency. Hence, the GAF-SMART presents the IoT adaptive hierarchical scheduling model to manage the series of diverse IoT data. The proposed task scheduling model focuses on several objectives that involve accomplishing efficiency while ensuring high resource utilization and providing Quality of Service (QoS) to the end-users.

Figure 1 shows the proposed GAF-SMART architecture of IoT adaptive mobile cloud computing. The GAF-SMART approach employs the cooperative Gaming model-based Fuzzy logic to optimally execute the IoT mobile applications, which incorporate three phases such as IoT mobile application workflow model, model the task scheduling as a hierarchy and IoT adaptive resource allocation. In the First, the GAF-SMART aims at defining a specific IoT mobile application workflow model according to the request in the cloud environment. In the second, the GAF-SMART identifies the corresponding application logic and workflow of the application. It selects the appropriate stream data from the plethora of diverse data available using gaming model-based fuzzy logic. It defines the hierarchical structure of the associated tasks based on the application logic, which results in the task prioritization of IoT application. The layered hierarchical structure enables the mobile cloud IoT system to perform the preceding task and consequently execute its succeeding tasks of an application without violating the application workflow. In the third, the game theory based adaptive task scheduler optimally allocates the cloud resources based on the local and the global best-fit solutions while emphasizing the reliability and efficiency of the mobile cloud services.

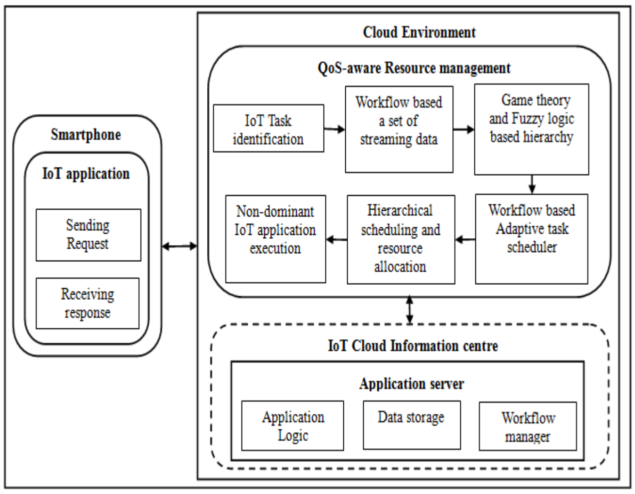
Cloud Environment
QoS-aware Resource management
IoT Task identification
Workflow based a set of streaming data
Game theory and Fuzzy logic based hierarchy
Non-dominant IoT application execution
Hierarchical scheduling and resource allocation
Workflow based Adaptive task scheduler
Smartphone
IoT application
Sending Request
Receiving response
IoT Cloud Information centre
Application server
Application Logic
Data storage
Workflow manager

Figure.1 GAF-SMART architecture of IoT adaptive mobile cloud computing

3.1 IoT mobile application workflow model

The GAF-SMART presents the framework for cloud-based IoT mobile applications. Most of the real-time IoT applications are time-bounded. IoT systems often require providing timely responses to the streaming data, which are likely to involve stream processing. For instance, the GAF-SMART takes into account the application of car parking system in the IoT-based smart city environment. When the mobile cloud user who has the car parking mobile application submits the car parking application request, the system analyzes the context of the user’s request as well as the corresponding streaming data to provide the best car parking lot. A car parking system in a smart city environment utilizes the different sensing technologies such as Radio Frequency Identification (RFID) for detecting the car parking access control. The laser, microwave radar, passive infrared, passive acoustic array sensors, Closed Circuit Television (CCTV) and ultrasonic for identifying the car parking lots and so on. To provide the response to the best car parking lots for car parking system in a smart city, the system needs to distinguish the different car parking areas such as a residential area, a transportation hub area, a shopping mall area and a street area. Each car parking area has the deployed sensors which periodically send the information about the utilization of the car parking lots to the IoT cloud information center.

The GAF-SMART essentially needs the prerequisite process of identifying the context of the IoT mobile application in order to leverage the application execution in the cloud environment. It determines the application logic in terms of the relevant streams of the general purpose streaming data requests, in which each task relies on its associated tasks. The GAF-SMART supports many unprecedented mobile applications by enabling the heterogeneous types of IoT requests in the mobile cloud infrastructure. It employs the IoT cloud information center including application logic, data storage, and workflow model while scheduling the tasks.

3.2 Modeling task scheduling as a hierarchy

The GAF-SMART exploits the Cloudlet architecture [17] to run the resource-intensive mobile applications in the remote server in which the mobile devices act as a thin client. The Cloudlet architecture enables the GAF-SMART to offload the computation to the nearby cloud infrastructure instead of offloading to the distant clouds, which eliminates the latency incurred by the wide-area network. It also supports the parallelism of computation offloading and real-time analytics of IoT application. The succeeding tasks of an application are retrieved from the IoT cloud information center containing the application logic along with the relevant stored data. Before the selection of the appropriate cloud resource for each task, the GAF-SMART needs to obtain the cloud tasks of an IoT application from the offloading manager and identify the hierarchical model of the received tasks.

3.2.1 Defining hierarchies and providing priorities to the tasks

To hierarchically model the tasks, the proposed adaptive task scheduler employs the game based fuzzy logic concepts on the arrival of offloaded tasks. Fuzzy pattern recognition method is based on the IoT application workflow model and exploits the fuzzy sets to describe the application pattern based on the principle of maximum membership degree. The main aim of the fuzzy logic is to determine the mapping relation of task and application pattern based succeeding tasks, which stabilizes the task levelling results and leads the robust results.

In a mobile cloud environment, let 'm' be the tasks and '*n*' be the application patterns. A set of cloud tasks (C_T) are defined as, $T=\{t_1,t_2,...,t_i,...,t_m\}$, and a set of application patterns is represented as $A=\{a_1,a_2,...,a_j,...,a_n\}$. Workflow of hierarchical level application logic is denoted as, $a_j=\{h_1^j, h_2^j,...,h_k^j,...,h_w^j\}$ and each cloud task (t_i) has a set of succeeding tasks to complete the application execution, $t_i=\{t_1^i,t_2^i,...,t_s^i,...,t_l^i\}$.

Let, $T \in C_T$; $C_T(t_i) \subseteq A_M^j$; $a_j \subset A$; $h_k^j \subset a_j$; $t_s^i \subseteq T(t_i)$

The hierarchy level of each application varies according to the application logic. Application logic facilitates the mapping of task (t_i) with its corresponding application (a_j), which is used to determine the hierarchical levels of application. Hierarchical level (h_k^j) of each application (a_j) is denoted as the succeeding task (t_s^i) of the respective task (t_i), which is represented by equation (1).

$$a_j \rightarrow h_k^j \rightarrow \{\{A\},\{B\},\ldots,\{Z\}\};\ A(t_i) \in T(A(t_i)) \quad (1)$$

The GAF-SMART maps a set of tasks with various application logic and finds the hierarchy level tasks for cloud task of a mobile application (A_M^j) according to the application workflow. Task-application mapping is based on the task type ($Task_{type}$) representing the main task needs to be accomplished in the application workflow like IoT application.

To map the main task with the appropriate application logic, fuzzy pattern recognition method analyzes the task input parameters ($IP(t_i)$) include *task ID*, $Task_{type}$, and Context of a task ($C(t_i)$) referring exact processing demands, which is shown in equation (2).

$$IP(t_i) \rightarrow \{task\ ID,\ Task_{type},\ C\} \quad (2)$$

Workflow application model for IoT application dynamically adopts the idea of hierarchy. When a new cloud task arrives, only the application pattern that is corresponding to the task level leads the dynamic hierarchical model. Equation (3) represents the task application mapping based on the context.

$$\{t_i,a_j\} \leftarrow \left(Task_{type}(t_i)=a_j\ \&\ C(t_i)=C(a_j)\right) \quad (3)$$

The workflow model describes the structure of an application process comprising a set of sequences. The proposed framework triggers each task request to a corresponding workflow execution process. The application pattern facilitates the system to automatically determine the succeeding tasks of each cloud task after receiving the task as the preceding task from the mobile user in which each succeeding task has the different characteristics and workflow nature. According to the workflow of the application and Best Fit (*BF*) of the resource (R_j), the distinct value of succeeding task (t_s^i) is acquired from a set of streaming data ($\mathcal{T}(t_i)$) relevant to the task (t_i) using equation (4).

$$\left(\mathcal{T}\left(A(t_i)\right)\right)=\{A(t_i)_1,A(t_i)_2,..,\mathrm{A}(t_i)_n\};BF$$
$$\left(\mathcal{T}\left(A(t_i)\right)\right)\rightarrow A(t_s^i) \tag{4}$$

In equation (5), the BF policy of a set of stream data on the cloud resources depends on the selection of lowest availability of resources and optimal execution time of each stream data.

$$t_s^i=\left\{\begin{matrix}\left(BF\left(\mathcal{T}\left(A(t_i)\right)\right)\text{-}R_j\right),\left(BF\left(\mathcal{T}\left(B(t_i)\right)\right)\text{-}R_j\right),\dots,\\ \left(BF\left(\mathcal{T}\left(Z(t_i)\right)\right)\text{-}R_j\right)\end{matrix}\right\} \tag{5}$$

To determine the succeeding task from a set of stream data, the GAF-SMART exploits the fuzzy model and game-theory based resource allocation method. A fuzzy model is a rule-based method involving a set of fuzzy rules. Fuzzy logic consists of fuzzy sets with associated membership functions. In this case, obtaining the relevant data as the succeeding task from a set of stream data is referred as the fuzzy set. The membership function indicates the uncertain value of stream data while accomplishing the objective factor of succeeding tasks. The following equation (6) is used to measure the scheduling probability of each task on the cloud resources.

$$t_s^i\rightarrow SP_i^j=\left(Task_{type}(t_i)=a_j\ \&\ C(t_i)=C\left(a_j\right)\&\ BF\left(\mathcal{T}(t_i)\text{-}R_j\right)\right) \tag{6}$$

According to the game-theory and objective factors, the GAF-SMART obtains the solution of the fuzzy set by optimally allocating the stream data on the VM resources. The objective factors of identifying the fuzzy set are based on the application logic, required data of processing a specific application, and workflow of an application. From each set of streaming data, the proposed system generates each task as the succeeding task for the primary task, which results in a set of succeeding tasks. It reveals that a set of succeeding task indicates a set of hierarchy levels which are used to perform the workflow based task scheduling progressively. In the succeeding task generation phase, the term ‘predecessor’ and ‘successor’ represents the preceding and succeeding stream data in a specific set of stream data source.

$$BF\left(\mathcal{T}(t_i)\text{-}R_j\right)\subseteq\begin{pmatrix}\mathcal{T}\left(A(t_i)\text{-}R_j\right)\mathcal{T}\left(B(t_i)\text{-}R_j\right)\&,\dots,\\ \mathcal{T}\left(Z(t_i)\text{-}R_j\right)\end{pmatrix} \tag{7}$$

Equation (7) denotes that the best-fit solution to each request depends on its succeeding tasks. The GAF-SMART initially forms the coalition consisting a set of best-fit resources of each stream data for each task (t_i) by considering the task parameters, application logic and optimal processing capability of resource characteristics in Non-Cooperative (*NC*) manner, which is known as a local best-fit solution (BF_L) using the equation (8).

$$BF_L\left(\mathcal{T}(t_i)\text{-}R_j\right)=<P(t_i)\left(a_j\right),CPU\left(R_j\right),Memory\left(R_j\right),Cost\left(R_j\right)>$$
$$Coalition\rightarrow\left\{BF_L\left(\mathcal{T}(t_i)\text{-}R_j\right)\right\} \tag{8}$$

Also, it focuses on the dominance factor (*D*) and local best-fit solution parameters to obtain the global best-fit solution (BF_G), which delivers a set of succeeding tasks (t_s^i) for each task (t_i) using the equation (9).

$$BF_G\left(\mathcal{T}(t_i)\text{-}R_j\right)=\langle D,BF_L\left(\mathcal{T}(t_i)\text{-}R_j\right)\rangle$$
$$SP_i^j\rightarrow\left\{BF_G\left(\mathcal{T}(t_i)\text{-}R_j\right)\right\} \tag{9}$$

Figure 2 illustrates the gaming model for fuzzy based IoT mobile application execution in the cloud. For instance, consider a mobile user submits a car parking application request to the cloud to receive the response as the car parking lot. The IoT cloud information center has a set of application logic along with its workflow. When a mobile user offloads a request (t_i) to remote server, the GAF-SMART exploits the pool of application logic to determine the corresponding application logic and its workflow. $IP(t_i)$ includes a *task ID* as the car parking application, $Task_{type}$ as the IoT application, and $C(t_i)$ as either optimal car parking lot or car parking availability areas.

Consider, the workflow of a specific car parking application includes three hierarchy levels of *A*, *B*, and *C* in which each level has a set of stream data storage in the cloud. *A*, *B*, and *C* are the location sensor, ultrasonic sensor, and Infrared sensor respectively for a car parking application. Succeeding tasks like $A(t_1^i)$, $A(B)(t_2^i)$, and $A(B)(C)(t_3^i)$ are obtained from a set of stream sources of *A*, *B*, and *C* respectively. The result of

$A(B)(t_2^i)$ depends on the result of $A(t_1^i)$ and the $A(B)(C)(t_3^i)$ result depends on the result of $A(B)(t_2^i)$. Then, the succeeding tasks as the hierarchy levels are given to the proposed algorithm of combined fuzzy logic and game-theory based resource allocation algorithm to execute the complete workflow of the car parking application and receive the available car parking lot response (t_i^R) nearby the location of the mobile user.

3.3 IoT adaptive resource allocation in mobile cloud

In the proposed game-theory method, each task and its succeeding tasks need to identify the appropriate scheduling plan on the cloud resource. Accordingly, the resource allocation is performed in two stages that involves succeeding task identification and optimal application execution in the cloudlet. It exploits the player strategies and objective factors as the fuzzy sets and fuzzy rules. The method takes into account the two sets of players, including a set of tasks (t_i, t_s^i) of an application (A_M^j) and a set of available resources (R_j) for a specific task. Also, it includes the objective factors such as execution time, resource utilization, including CPU, memory and energy cost.

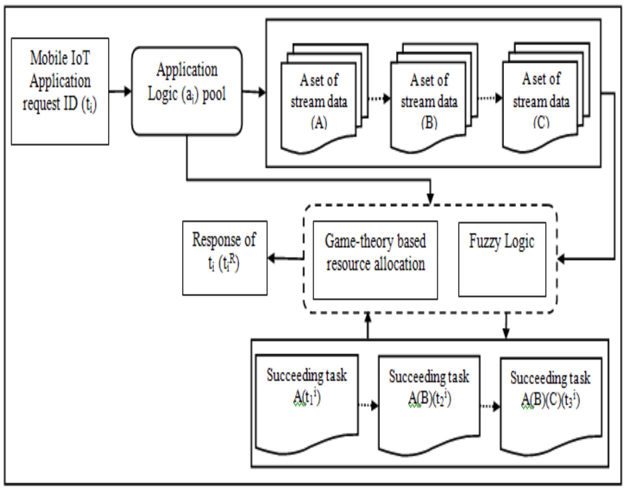
Mobile IoT Application request ID (t_i)
Application Logic (a_i) pool
A set of stream data (A)
A set of stream data (B)
A set of stream data (C)
Response of t_i (t_i^R)
Game-theory based resource allocation
Fuzzy Logic
Succeeding task A(t_1^i)
Succeeding task A(B)(t_2^i)
Succeeding task A(B)(C)(t_3^i)

Figure.2 Gaming model for Fuzzy based IoT mobile application execution in the cloud

Equation (10) identifies an optimal set of resources based on the succeeding tasks of each task and the availability of the computing and storage resources, which is the local best-fit solution (BF_L). This coalition formation is in proximity to the optimal solution for energy-efficient application execution.

$$BF_L\left((t_i\&\, t_s^i)\text{-}R_j\right)=\langle ET_{ij},U,Cost\rangle$$

$$Coalition\rightarrow BF_L=\left\{BF\left(\{t_i,R_j\}\&\{t_s^i,R_j\}\right)\right\} \qquad (10)$$

The GAF-SMART approach focuses on both the self-performance of each task that refers the execution time and the cooperative performance of an application to obtain the optimal results in minimum makespan precisely. Hence, it initially determines the resources in non-cooperative manner and then, finds the resources in a cooperative manner if only dominance occurs. Using equation (11), the global optimal solution is obtained for allocating the tasks of an application to the cloud resources.

$$BF_G\left((t_i\&\, t_s^i)\text{-}R_j\right)=\langle M_{ij},D,BF_L(t_i\&t_s^i)\rangle$$
$$SP_i^j\rightarrow\left\{BF_G\left((t_i\&\, t_s^i)\text{-}R_j\right)\right\} \qquad (11)$$

The local best-fit solution is likely to extend the application completion time due to the dominance of optimal solution of one task on other tasks of the same mobile application. Hence, the GAF-SMART simultaneously maps the makespan and also the availability of the appropriate resources. The identification of the dominant task depends on the maximum value of a task while comparing other tasks of an application. The maximum value is ($D>avg(x)+a$) the average value of all tasks of an application in which 'a' depends on the loading time from one task to another task of an application, where $0\leq a\leq 1$. After selecting a set of optimal resources, the adaptive task scheduler assigns each schedulable task to the appropriate resource by maintaining application workflow and using the global BF policy. In the game-theory based resource allocation model, the fuzzy sets and fuzzy rules based scheduling plan initially prefers a set of resources having the following requirements:

$$SP_i^j=\begin{cases} 1;\ \left(\text{if } M_{ij}=low\ \&\ C_{ij}=low\right)\text{or}\left(\text{if } U_{ij}=high\ \&\ C_{ij}=low\right)\\ 0;\ \left(\text{if } M_{ij}=high\ \&\ C_{ij}=high\right)\text{or}\left(\text{if } U_{ij}=high\ \&\ C_{ij}=high\right)\end{cases} \qquad (12)$$

In equation (12), '1' and '0' refers the applicable and non-applicable resources for a specific application. The global BF policy especially focuses on the dominance value of each task's optimal solution. The dominance value refers that the execution time of a single task may extend the waiting time of its succeeding task on a particular resource, resulting dominance value is '1', otherwise '0'. If the dominance value occurs, the GAF-SMART needs to optimally utilize the available resources regardless of violating the application workflow. By exploiting the global BF policy on the cloud resources, the GAF-SMART assures both the mobile user and the cloud service provider offering the high QoS and ensuring the minimum energy cost respectively. In this case, the BF policy is based on the selection of lowest availability of resources and negligible dominance value of one task on others. It is done by cooperatively selecting the optimal resource for the dominant task or preceding task within the coalition resources using equation (13). It also satisfies the maximum utilization of each resource while hierarchically scheduling the tasks based on the estimated makespan (*M*).

$$SP_i^j = \begin{cases} \left(M_{ij}\right)^{-1} * \sum_{i=1}^{I} \left(\frac{U\left(a(t_i)\right)_{cj}}{B\left(R_{cj}\right)}\right) & \text{, if } BF\left(t_{ij}\right)=NC \\ \left(ET_D\left(P(t_i)\right)_{cj} - ET_N\left(P(t_i)\right)_{cj}\right) * \left(\frac{U_N\left(P(t_i)\right)_{cj}}{B\left(R_{cj}\right)}\right), & \\ \text{if } BF_L\left(P(t_i)\right) > BF_L\left(S(t_i)\right) \text{ or } BF_l(t_i)=D & \end{cases} \tag{13}$$

Where, $U(a(t_i))_{cj}$ denotes the utilization of jth resource in cth coalition after allocating the all ith task of an application (a) in the corresponding resource. $B(R_{cj})$ is the jth resource availability in the cth coalition. $BF_L(P(t_i))$ and $BF_L(S(t_i))$ represents the local optimal solution of the preceding task and succeeding task respectively. $ET_D(P(t_i))_{cj}$ refers the execution time of predecessor ($P(t_i)$) which delays the start time of its successor ($S(t_i)$). $ET_D(P(t_i))_{cj}$ and $ET_N(P(t_i))_{cj}$ denote the dominant execution time of the preceding task on a specific coalition resource (cj) and novel execution time of the preceding task on all others coalition resource (cj). $U_N(P(t_i))_{cj}$ is the utilization of predecessor on all the coalition resource (cj) excluding the resource providing the dominance result for that specific predecessor. Using equation (13), the GAF-SMART schedules the ith task on the jth cloud resources based on the higher weight of scheduling plan on a particular resource. Thus, the GAF-SMART manipulates the cloud resources while executing the IoT mobile application and provides the energy-efficient cloud services.

4. Results and Discussions

To illustrate the experimental results, the effectiveness of the GAF-SMART technique is compared with conventional Nested Game based offloading method for MCIoT system (NG-MCIoT) system [18].

4.1 Experimental setup

The proposed evaluation framework implements and validates the GAF-SMART system through a simulation model of WorkflowSim tool. It considers that the input application is the cloud-based IoT mobile application in which mobile device handles several task processing and the remote server manipulates the intensive tasks of the IoT mobile application. The proposed experimental model runs the experiments on Linux Ubuntu 12.04 LTS 64-bit machine, 2.9 GHz Intel CPU and 32 GB memory. The experimental setup considers 10 hosts and 30 VMs, and Million Instructions Per Second (MIPS) of VMs varies from 800 to 1000 MIPS. It considers the mobile device energy consumption as 1kWh and thus, the device consumes 0.27J/s.

4.1.1 Evaluation metrics

Makespan: It is the overall completion time of a mobile application, including the execution of tasks.
Average energy consumption: It is defined as the average of total energy consumed by the mobile devices while executing the mobile applications until completion of execution.
Response time: It is the interval between the service initiated time of an application and service end time of that application by the cloud service provider.

4.2 Experimental results

The experimental results illustrate the performance of the GAF-SMART in terms of makespan, average energy consumption and response time by assessing the proposed system in various scenarios. The GAF-SMART approach outperforms the existing NG-MCIoT approach in terms of maintaining the trade-off between QoS and operational cost. Since the NG-MCIoT approach assigns the deadline for each task based on the application and consequently, selects the cloud resource based on the minimal price and deadline. It requires the repeated dynamic auction procedure to determine the mobile cloud users' desire with the request characteristics and makes the desired price decision for the similar kind of users in the next auction. Application based deadline assignment and the resource price based resource selection does not provide better results in all the cases of IoT mobile applications. It fails in offering the minimal energy consumption and the optimal makespan when increasing the number of streaming data. To tackle this constraint, the GAF-SMART exploits the workflow model based adaptive task scheduler to maintain both the effective resource utilization and makespan. Moreover, it employs the game theory model and fuzzy logic to cooperatively determine the optimal solution while ensuring the non-dominant task execution.

4.2.1 Makespan Vs Number of VM instances

Figure 3 illustrates the makespan of both the GAF-SMART and NG-MCIoT approach while varying the number of VM instances and the Amount of Load Change (ALC). ALC refers to the sudden load changes of 10%, 20%, and 30%, that depends on the arrival of the requests to the resources. The impact of increasing the number of VM instances reflects that there is a linear decrease of makespan. When ALC=10%, the GAF-SMART approach suddenly reduces the makespan value by 20.93% while varying the number of VM instances from 5 to 25 but, the makespan of NG-MCIoT approach reduces by 9.79%. The GAF-SMART approach offers the service with optimal makespan in the range of up to 23.99% variation even while dynamically changing a large number of loads from 10% to 30% when the number of allocated VM instances is 25 on the remote server. At the same scenario, the existing NG-MCIoT approach obtains the 25.31% of makespan variation, which creates the negative impact on resource utilization and also the QoS due to the consideration of price and deadline satisfaction in all the scenarios. The GAF-SMART system optimally utilizes the VM instances and provides an optimal solution based on the cooperative game theory model and fuzzy logic.

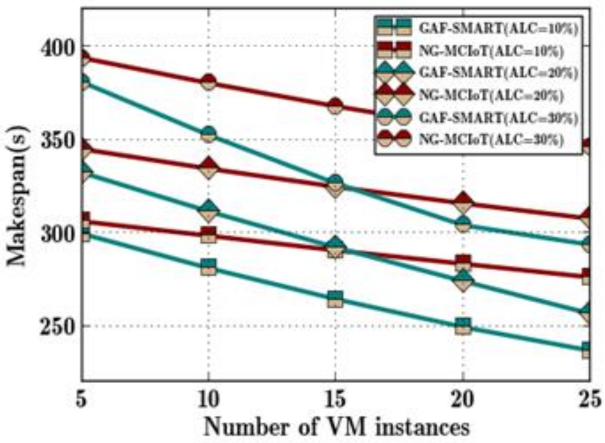

Figure.3 Makespan Vs Number of instances

4.2.2 Average energy consumption Vs Percentage of load

Figure 4 depicts the comparative results of both the GAF-SMART and NG-MCIoT approaches while increasing the percentage of the load from 10% to 50% and the Average Task Length (ATL) of an application from 25KB to 100 KB. It reveals that the average energy consumption of the mobile devices significantly increases when increasing the ATL from 25KB to 100 KB. The GAF-SMART approach leads the device to consume the energy 40.91% while increasing the load from 10% to 50%, however, in the NG-MCIoT system, the device consumes 60.23% when the ATL is 25KB. It is because the GAF-SMART approach focuses on the hierarchical task scheduling method that greatly mitigates the energy consumption and preserves the device energy during IoT application execution. At the point of 30% load and ATL is 50KB, the NG-MCIoT approach reduces the energy consumption by 4.05% than the GAF-SMART, but the GAF-SMART approach gradually decreases the energy consumption even a percentage of the load increases. The existing system fails to deliver better performance when suddenly changing the load since the price and deadline factors are likely to extend the completion time when there is an uncertainty of deadline.

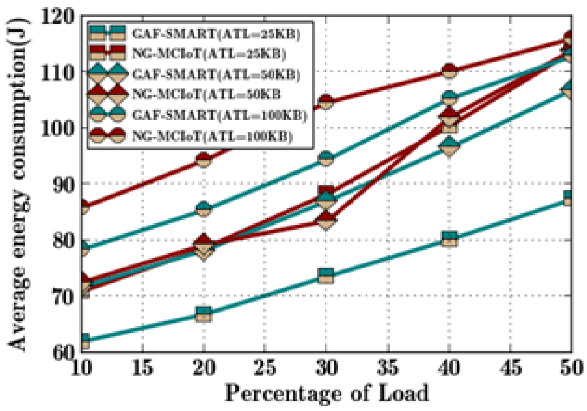

Figure.4 Average energy consumption Vs Percentage of load

4.2.3 Response time Vs Average task length

Figure 5 demonstrates the response time of both the GAF-SMART and NG-MCIoT while varying the average task length from 25 KB to 125 KB. An increasing average task length of an application creates the impact of prolonging the response time due to the growing workload of the task size. The GAF-SMART approach reasonably extends the response time by 11.9% when increasing the average task length from 50KB to 125 KB. However, the existing NG-MCIoT approach drastically escalates the response time by 23.45% as it does not consider the Cloudlet architecture for offloading and workflow based cooperative execution of tasks for a specific application. When the average task length is 50KB, the NG-MCIoT system suddenly reduces the response time and then, rapidly increases the response time than the GAF-SMART. The GAF-SMART approach focuses on the dominance factor throughout the application execution. Hence, it maintains the optimal response time. Moreover, the GAF-SMART approach precisely determines the appropriate succeeding tasks from a set of streaming data and manipulates the tasks in a hierarchical manner to further perform task scheduling.

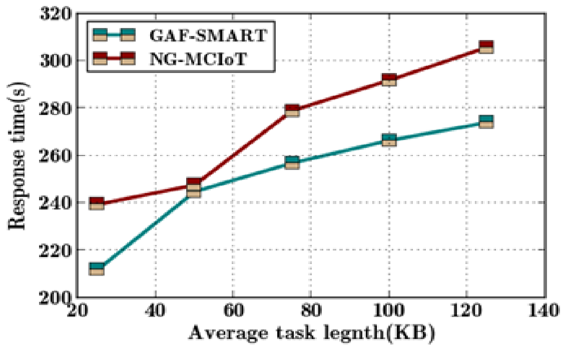

Figure.5 Response time Vs Average task length

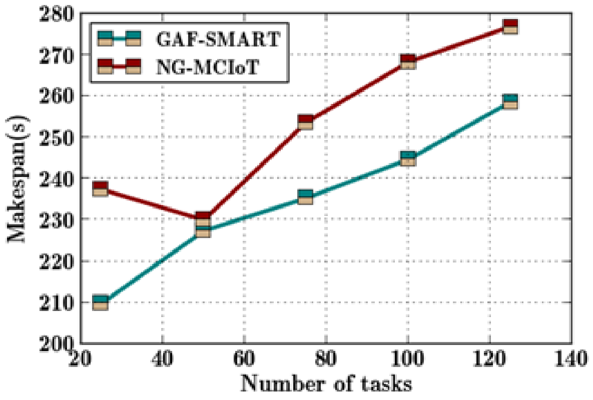

Figure.6 Makespan Vs Number of tasks

4.2.4 Makespan Vs Number of tasks

Figure 6 shows the makespan of GAF-SMART and NG-MCIoT approaches with the impact of the number of tasks varied from 25 to 125. The makespan linearly raises when increasing the number of tasks per application. The GAF-SMART approach increases the makespan value marginally by 13.72% even increasing the number of tasks from 25 to 125. The GAF-SMART approach targets on providing the non-dominaent execution of tasks within the application while maintaining the hierarchical application workflow and cooperative execution of tasks. On the other hand, in the same scenario, the NG-MCIoT approach escalates the makespan by 20.34%. After increasing the number of tasks from 50, the NG-MCIoT system suddenly decreases the makespan value due to the dominance regardless application execution of the NG-MCIoT system. The GAF-SMART approach employs the Cloudlet architecture, especially in the context of IoT mobile application, which facilitates the delay-aware execution of the mobile application.

5. Conclusion

This work proposes a novel mobile cloud IoT paradigm for executing the real-time IoT applications with the support of Cloudlet architecture in the mobile cloud environment. The proposed GAF-SMART focuses on the workflow of an application and adaptive application execution while ensuring a minimum level of energy consumption both in the device and the remote server by exploiting the adaptive task scheduler. It applies the game theory model-based fuzzy logic on the two stages involving the identification of succeeding tasks from numerous data stream and the execution of IoT adaptive application in a hierarchical manner. Furthermore, the mobile cloud IoT paradigm is capable of automatically defining the allocation of hierarchically scheduled tasks based on the non-dominance execution. The GAF-SMART focuses on the cooperative execution of an application to minimize the makespan and enhances the resource utilization, resulting in the energy-efficient system. The experimental results of the GAF-SMART outperform the existing NG-MCIoT approach by accomplishing QoS in terms of makespan by 17.86%. The GAF-SMART approach considers the workflow of IoT application throughout the execution as well as the non-dominant execution time even achieving better resource utilization. Moreover, it effectively obtains its prerequisite process of succeeding task identification of the numerous streaming data using the game-theory model along with the fuzzy logic.

The future work focuses on the different IoT mobile application execution workload across the multiple cloud data centers and the VM migration during application execution in the distributed environment to further improve the performance of the proposed system.

References

[1] N. Fernando, S.W. Loke, and W. Rahayu, “Mobile cloud computing: A survey”, *Elsevier transaction on Future Generation Computer Systems*, Vol.29, No.1, pp. 84-106, 2013.

[2] H.T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches”, *Wireless communications and Mobile Computing*, Vol.13, No.18, pp. 1587-1611, 2013.

[3] K. Kumar, J. Liu, Y.H Lu, and B. Bhargava, “A survey of computation offloading for mobile

systems", *Springer Transaction on Mobile Networks and Applications*, Vol.18, No.1, pp. 129-140, 2013.

[4] D. Singh, G. Tripathi, and A.J Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services", *IEEE world forum on Internet of things (WF-IoT)*, pp. 287-292, 2014.

[5] A. Botta, W. De Donato, V.Persico, and A. Pescapé, "On the integration of cloud computing and internet of things", *IEEE International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 23-30, 2014.

[6] T. Shon, J. Cho, K. Han, and H. Choi, "Towards advanced mobile cloud computing for the internet of things: current issues and future direction", *Springer transaction on Mobile Networks and Applications*, Vol.19, No.3, pp. 404-413, 2014.

[7] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing", *ACM SIGMETRICS Performance Evaluation Review*, Vol.40, No.4, pp. 23-32, 2013.

[8] Liu, Yanchen, and Myung J.Lee, "An effective dynamic programming offloading algorithm in Mobile cloud computing system", *IEEE Transaction on Wireless Communications and Networking Conference (WCNC)*, pp. 1868-1873, 2014.

[9] H. Flores, and S. Srirama, "Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning", *ACM Proceeding of the fourth workshop on Mobile cloud computing and services*, pp. 9-16, 2013.

[10] Yibin Li, Min Chen, Wenyun Dai, and Meikang Qiu, "Energy optimization with dynamic task scheduling mobile cloud computing", *IEEE Transaction on System Journal*, No.99, pp. 1-10, 2015.

[11] L. Shakkeera, and L. Tamilselvan, "Energy-Aware Application Scheduling and Consolidation in Mobile Cloud Computing with Load Balancing", *Springer transaction on Emerging Research in Computing, Information, Communication and Applications*, pp. 253-264, 2016.

[12] M.H. Zarei, M.A. Shirsavar, and N. Yazdani, "A QoS-aware task allocation model for mobile cloud computing", *IEEE Second International Conference on Web Research (ICWR)*, pp. 43-47, 2016.

[13] Yang Ge, Yukan Zhang, Qinru Qiu, and Yung-Hsiang Lu, "A game theoretic resource allocation for overall energy minimization in Mobile cloud computing system", *In Proceedings of ACM/IEEE International symposium on Low power electronics and design*, pp. 279-284, 2012.

[14] Zhanlin Ji, Ivan Ganchev, Mairtin O'Droma, Li Zhao, and Xueji Zhang, "A cloud-based car parking middleware for IoT-based smart cities: design and implementation", *Sensors*, Vol.14, No.12, pp. 22372-22393, 2014.

[15] S.S. Yau, and A.B. Buduru, "Intelligent planning for developing mobile IoT applications using cloud systems", *IEEE International Conference on Mobile Services*, pp. 55-62, 2014.

[16] S. Kim, "Nested game-based computation offloading scheme for Mobile Cloud IoT systems", *Springer EURASIP Journal on Wireless Communications and Networking*, No.1, p. 1-11, 2015.

[17] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing", *IEEE Pervasive Computing* Vol.8, pp.14–23, 2009.