Journal of Theoretical and Applied Information Technology
15th June 2017. Vol.95. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SATISFYING SLA OBJECTIVES OF SEAMLESS EXECUTION OF MOBILE APPLICATIONS IN CLOUD WITH NET PROFIT

[1]L.SHAKKEERA, [2]LATHA TAMILSELVAN

[1,2] School of Computer, Information and Mathematical Sciences, Department of Information Technology
B.S.Abdur Rahman University, Chennai, Tamilnadu, India.

E-mail: [1]shakkeera841@gmail.com, [2]latha.tamil@bsauniv.ac.in

ABSTRACT

Mobile Cloud Computing (MCC) technology enables the resource-constrained Smartphones to outsource its richer applications onto resource-abundant cloud. Cloud computing has emerged as a platform to extend the capabilities of mobile devices regarding energy and resources. MCC offers the services to enhance mobile user convenience, the energy-hungry applications require selective offloading of tasks onto the cloud and optimally scheduling the tasks by taking into account the cloud resources. Even though, the MCC offers the services with high level user convenience, it creates numerous challenges for the service providers. This paper proposes Maximizing User anTicipation on cloUd-based mobile AppLications BesidE NEt proFIT (MUTUAL-BENEFIT), an optimization approach that satisfies the Service Level Agreements (SLA) of the users and simultaneously maximizes the profit of the service provider. The MUTUAL-BENEFIT significantly enhances task offloading, scheduling and resource allocation in MCC environment. Initially, offloading manager of ThinkAir architecture judiciously offloads the intensive tasks based on the knowledge of device energy and performs parallelized execution. Secondly, the non-recursive dynamic programming assisted Ant Colony Optimization (ACO) method schedules the intensive tasks for an extended battery and quick response. It contemplates the SLA objective factors such as execution time, load and profit as the pheromone value of ACO while selecting the optimal resources. Together with SLA-aware task scheduling and the utility function based optimal VM resource allocation maintains an excellent trade-off between the service satisfaction and profit. The experimental results show that MUTUAL-BENEFIT approach preserves the device energy while ensuring the optimal SLA objectives and profit of the provider.

Keywords: *Mobile Cloud Computing, Resource Allocation, Dynamic Programming, Ant Colony Optimization, Energy aware, Load Balancing, Profit.*

1. INTRODUCTION

In recent years, due to the rapid growth of the sophisticated mobile applications, the mobile computing has become vigorously popular in the pervasive environment. Unfortunately, the mobile devices face the resource scarcity problem regarding a short battery lifetime due to the resource intensive mobile applications. Therefore, many of the sophisticated applications are not suitable for the resource-poor mobile devices. The major constraint is the resource-intensive mobile applications run on resource-hungry mobile devices. To resolve this constraint, enhancing the hardware and software on the mobile device is a necessity. The hardware changes on the mobile device are unable to resolve the energy constraints. As a result, a cloud platform is essential to enable the mobile device with the energy efficient execution. In Mobile Cloud Computing (MCC), the Smartphone has two major constraints such as battery lifetime and response time of the mobile user. In general, the offloading method saves the mobile device energy, but mobile users still face energy as a major constraint while executing the resource-rich applications on mobile devices. Thus, energy saving is considered as a prime factor while offloading and executing the resource-rich mobile applications. The mobile devices save the energy if the remote server executes a maximum portion of an application. The mobile devices have to run

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the basic functions of a mobile application because offloading of an entire mobile application to the cloud server is not possible. Energy saving depends on the optimal minimization of resource-rich application's execution time on a remote server. Hence, energy consumption of the mobile device depends on the overall completion time of the mobile application on a mobile device, transmission channel, and cloud server. Mobile Cloud Computing [1,2] augments capabilities of the mobile devices by utilizing the enormous cloud resources in an on-demand fashion. Remote processing related approaches execute the applications using virtualization technology under different mobile cloud architectures [3]. Service Level Agreement (SLA) is a crucial consideration of both the perspectives of the mobile end user and the cloud provider [4]. The service providers own the cloud data centers with the enormous computation and storage capabilities. In some cases, the service provider offers the over-provisioning of the resources while ensuring the SLAs or end user's response time. This over-provisioning leads to high resource cost for the service provider. Hence, the optimal resource allocation is imperative in MCC to reduce the end user's resource scarcity as well as the expenses of the cloud provider. Most of the conventional methods discuss the task scheduling and resource allocation on MCC either on the end user or cloud service provider. In MCC environment, millions of mobile users submit the same application request at the same time. Therefore, optimal scheduling and allocation [5,6] are important to make the significant impact on both the end user side and the provider side.

The cloud service provider makes SLA with the end user's requirements, where the specific, measurable characteristics of SLA are end user's mobile device energy and response time. The Maximizing User anTicipation on cloUd-based mobile AppLications BesidE NEt proFIT (MUTUAL-BENEFIT) approach optimizes offloading, task scheduling, and resource allocation in MCC environment to satisfy SLA of the end user and to enhance the profit of the provider. The research objective of the MUTUAL-BENEFIT approach is to provide the energy-efficient seamless mobile application execution without violating the SLAs. Moreover, it targets on maximizing the profit of the cloud service provider. In MUTUAL-BENEFIT, SLA objectives include minimum application completion time and energy consumption.

In brief, the contribution of the research paper is given below:

- The MCC prefers computation offloading mobile tasks to an external platform such as cloud and reduces the burden of the energy-constrained mobile device. Instead of dividing and executing the tasks recursively, the MUTUAL-BENEFIT approach exploits non-recursive dynamic programming in the ThinkAir architecture, where the resources are dynamically allocated to run the tasks in parallel. The ThinkAir architecture divides the application tasks into mobile and cloud tasks according to the mobile device energy.
- MUTUAL-BENEFIT approach divides the cloud tasks of similar applications into sub-tasks using non-recursive dynamic programming. The proposed approach schedules the tasks using ACO technique that finds the optimal VM resources with minimum computation time by exploiting dynamic programming. To satisfy the SLA objectives and optimally schedule the sub-tasks, it contemplates both the execution time and load as the pheromone value of ACO.
- Without violating the SLAs, the MUTUAL-BENEFIT approach enriches the cloud provider's profit by cut back surging resource costs in the cloud. It increases the net profit of the provider by judiciously distributing the tasks to VMs and allocating cloud resources to the dispatched tasks. Finally, it maintains the trade-off between the number of services with high quality and profit of service provider in terms of resource cost.

2. RELATED WORK

The previous research works investigate the offloading, task scheduling and resource allocation techniques to extend the battery life of the mobile devices. This section discusses the several conventional methods.

2.1. Task Scheduling

Energy and performance-aware (EAPA) task scheduling [7] resolves the minimal delay problem with the consideration of total completion time of the tasks using an initial task scheduling algorithm. It migrates the tasks for minimizing the device's energy using the rescheduling algorithm in a mobile cloud environment. An Energy-efficient, cooperative offloading model (E2COM) schedules the tasks with the minimization of mobile user energy consumption and internet data traffic redundancy in a mobile cloud environment [8]. The task scheduling algorithm depends on the pricing mechanism and Lyapunov optimization to reduce energy consumption in WLAN. Energy-optimal framework [9] solves scheduling problems of constrained optimization problem to minimize device energy. It focuses on both execution and transmission of mobile applications on the mobile device or cloud using the Lagrangian multiplier method. An effective distributed parallel scheduling [10] model develops a simulator for analyzing the bottleneck in terms of device energy and quality of the network in MCC.

A prognostic load balancing strategy in [11] minimizes the energy consumption of the mobile device and improves the response time and scalability. It reduces the latency of the system using the amplified-ESBWLC algorithm. Energy-aware workload placement model solves the multi-dimensional bin-packing problem (MDBP) using the nature-inspired algorithm. The dynamic placement of the workload depends on the current workload and ACO meta-heuristics [12]. VM resource scheduling [13] balances the load based on the genetic algorithm in a cloud environment. It considers the historical information and current load of the system while scheduling the resource. ACO based load balancing [14] considers the routing packets as the ants in the cloud environment. It replaces the routing tables with a probability value of pheromone tables which contains the information of pheromone value and incremental pheromone update.

2.2. Resource Allocation

A nested two-stage optimization framework [15] takes effective offloading decision for minimizing device energy and improving the response performance in the first stage. Cloud computing controller allocates the resources for increasing provider's profit in the second phase. An agent-based optimization framework [16] minimizes the requirements for cloud server using mobile-agent based partition. The mobile agents partition the application based on the device energy for taking an offloading decision. The ThinkAir in [17,18] dynamically allocates the resources using an execution controller of VM instances in the cloud. It provides the resources in an on-demand manner and execution controller takes optimal decision according to execution time, cost, and device energy. An adaptive computing resource allocation [19] maximizes the resource utilization and reward function of the system. The Semi-Markov decision process (SMDP) based resource allocation, achieves the objective function during consideration of both mobile device and cloud. Game-theoretic resource allocation framework [20] allocates the resources based on the device energy. Nash equilibrium algorithm solves energy minimization problem (congestion game) in a mobile cloud environment. Cloud-assisted motion estimation (CAME) [21] approach discusses the resource allocation of the cloud resources for mobile video compression and estimation. It saves the mobile device energy while processing the compute-intensive applications of mobile video streaming.

An MAPCloud [22] resolves the NP-hard optimization problem using simulated annealing based heuristic. It allocates the cloud resources for mobile applications based on the 2-Tier cloud architecture. Resource allocation and scheduling strategy maximizes the cloud provider's profit and satisfies SLA using dynamic rank based resource allocation and gi-FIFO scheduling. It considers the context-aware application requests in distributed cloud [23]. In multi-tier cloud computing environment, the force directed searching method based algorithm allocates the resources to maximize the total profit while meeting SLAs. It provides processing, memory and communication resources to the user based on the SLA [24]. Auction based mobile cloud approach [25] allocates the bundle of cloud resources to the mobile user and analyzes the incentive factors using auction mechanism. Dynamic programming based offloading algorithm (DPOA) [26] solves the optimization problem during offloading based on the optimal partitioning.

Most of the conventional methods focus on the energy based task scheduling and resource allocation in MCC environment. The realistic task scheduling model is necessary to maintain the trade-off between performance and cost. Hence, the proposed approach contemplates the SLA objectives and profit of the provider as the major constraints.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. SYSTEM MODEL

This section presents a system model for providing cloud services for the consideration of optimal scheduling and allocation. It is assumed that the mobile devices comprise of enormous processing capability if it outsources the resource-hungry applications to the cloud. MCC environment consists of a set of similar applications (A) from various mobile users i={1,2,...m}, and cloud resources j={1,2,....n}. An appropriate assignment of j ∈ cloud resources to A_i ∈ A provides the optimal service to the end-user 'i'. In cloud server, scheduling manager segregates the applications into tasks (T_i). To select the optimal VM for T_i, it is essential to consider the task completion time ($Ʈ_{ij}$), load balancing ($ŋ_{ij}(S_\omega'(t))$), and profit ($S_\omega(t)$) in which $ŋ_{ij}$ represents the optimal load balancing factor.

Infrastructure Service Provider: Infrastructure service provider (ISP) is known as the virtual resource provider. ISP provides the virtual resources in terms of Virtual Machines (VMs) to the cloud service provider (CSP). CSP rents the VMs to end-users based on the amount charged by the ISP. Each VM resource has unique configurations of CPU, price, and memory.

Cloud Service Provider: CSP is also known as the service provider. CSP provides the rented virtual resources to the end-users for processing mobile applications in the cloud. It selects the best S_P ∈ set of ISPs, and it furnishes the resources of S_P with execution services to improve user satisfaction level and its profit.

End-user: End-user must pay the fees to a service provider that depends on the SLA and received service utilization. The payment of End-user is the revenue of CSP. SLA violation reduces the revenue of A_i if the application takes longer time than average execution time. Thus, it is essential to consider both the energy cost and the revenue for maximizing the profit of the provider and satisfying the SLA objectives.

4. MUTUAL-BENEFIT METHODOLOGY

The MUTUAL-BENEFIT approach exploits optimal task scheduling and resource allocation process to execute the mobile cloud applications.

The MUTUAL-BENEFIT approach enabled mobile cloud environment ensures the seamless application execution resulting extending the battery lifetime and the optimal profit. The mobile cloud task scheduling and resource allocation process schedules the offloaded tasks and allocates the resources merely based on the availability and the resource requirements regardless of the MUTUAL-BENEFIT. The additional consideration of the proposed algorithm in mobile cloud environment facilitates both the mobile users and the providers in reducing the burden of application execution and mitigating the processing complexity respectively. The MUTUAL-BENEFIT creates the greater impact on tackling the battery constraint and manipulating dynamic numerous user requests with high profit. For instance, the Sudoku solver application contains a different number of cells based on the level of the application. The mobile device partially fills the cells in Sudoku solver application due to the energy constraint of the mobile device. The ThinkAir architecture based offloading manager monitors the energy model of the device to offload the resource-intensive tasks in partially filled cells of the Sudoku solver application to the cloud server. In Sudoku solver application, empty cells are considered as the cloud tasks. Non-recursive dynamic programming based ACO method schedules the cloud tasks by selecting the SLA objectives based optimal VM resources. Finally, the Bellman's theory based utility function optimally allocates the resources to determine the solution for empty cells. The selected optimal VM resources enable the corresponding task to execute the solution finding the process of the corresponding unfilled cells in Sudoku solver application. This optimal execution of MUTUAL-BENEFIT balances the objectives of both the end-user and the service provider. The proposed methodology of MUTUAL-BENEFIT is shown in Figure 1.

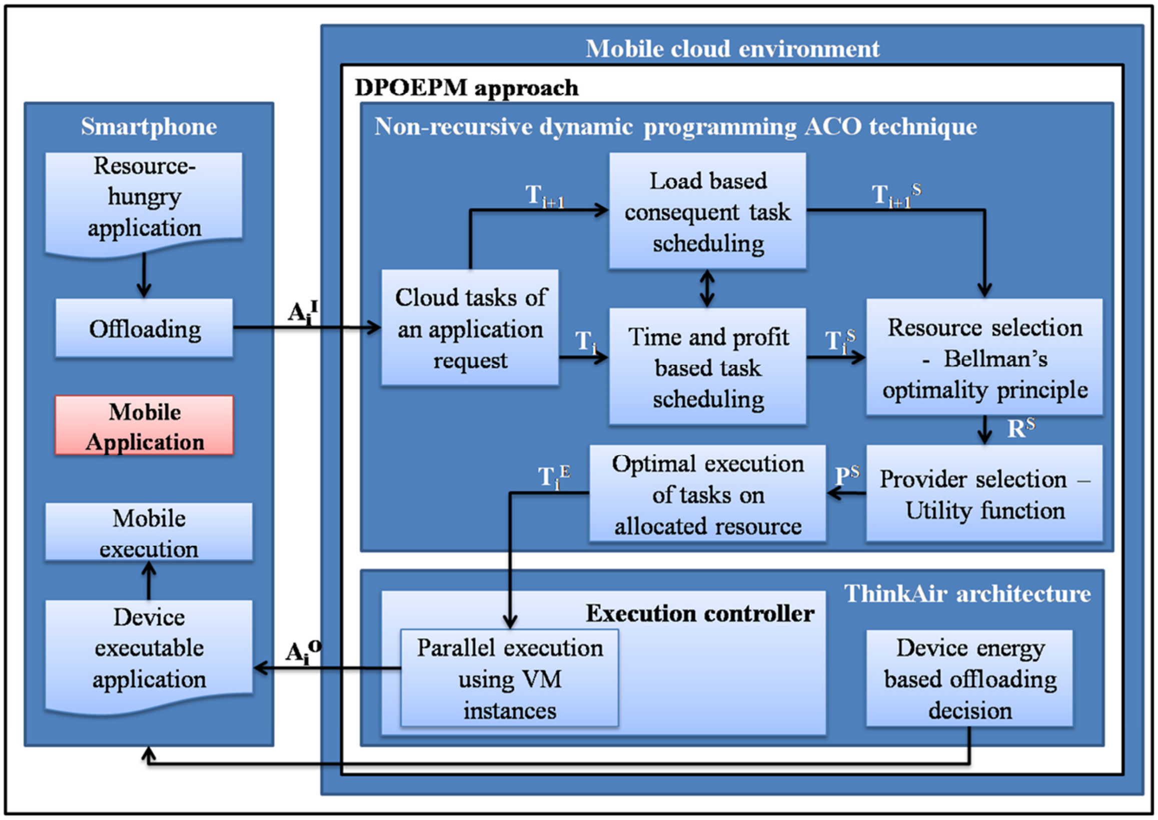

Figure 1. DPOEPM approach in MCC

4.1. Optimal Offloading Using the ThinkAir Architecture

The computation offloading aims to migrate the resource-intensive computations from a mobile device to the resource-rich cloud. It enhances the performance of mobile applications that are unable to execute in Smartphones due to insufficient battery energy resources. The MUTUAL-BENEFIT exploits the ThinkAir architecture to divide the application into mobile and cloud tasks according to the mobile device energy. The ThinkAir architecture enables the profiler to monitor the mobile device and retains the information for future offloading decisions. The profiler component of ThinkAir architecture monitors the profile information of the device, program, and network of the mobile device to identify the off-loadable tasks. Also, ThinkAir architecture supports to execute the dynamic programming in MUTUAL-BENEFIT, where the decision about recursive tasks is taken using the stored offloading information without re-execution. Further, it reduces the complexity of assigning tasks and finding optimal resources in MUTUAL-BENEFIT.

4.2. Satisfying SLA Objectives via Optimal Task Scheduling in the Cloud

The execution controller of ThinkAir architecture executes the MUTUAL-BENEFIT algorithm in a remote server. The main goal of the cloud provider is to satisfy the user convenience in service provisioning in terms of battery energy and response time. The MUTUAL-BENEFIT considers SLA as an important factor while performing task scheduling and resource allocation. These processes make an impact on the processing time of the remote server and the SLA objectives. Because, if the cloud manager delays the processing time of an application, it may violate the SLA constraints of battery usage and response time.

The MUTUAL-BENEFIT approach employs the ACO technique to schedule the tasks optimally and executes the non-recursive dynamic programming with the support of ThinkAir architecture. Before selecting the corresponding optimal cloud resources to the tasks using ACO technique, the MUTUAL-BENEFIT obtains the cloud tasks of an application from the offloading manager.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2.1. ACO technique

The MUTUAL-BENEFIT approach follows the basic function of ACO approach while identifying the best solution for task scheduling. The non-recursive dynamic programming ACO technique considers together of execution time, load balancing, and profit as Pheromone value to achieve SLA objectives. The Pheromone value of tasks in each VM is measured to achieve the optimal scheduling in MUTUAL-BENEFIT. Consider, the set (S_i) contains the cloud tasks, in which 'i' is varied from 1 to a number of cloud tasks. The pheromone value of each task (PV) is measured using the following equation (1).

$$PV_{ij}=\begin{cases} \dfrac{[\alpha(\tau_{ij}(t))]\,[S_\omega(t)_{best}]^\beta}{\sum_{n\epsilon allowed_\omega(t)}[\alpha(\tau_{in}(t))]\,[S_\omega(t)_{best}]^\beta} & \text{, if } T_i=T_1 \\ \dfrac{[\alpha(\tau_{ij}(t))]\,[(\eta_{ij})\,S_\omega'(t)]^\beta[S_\omega(t)_{best}]^\gamma}{\sum_{n\epsilon allowed_\omega(t)}[\alpha(\tau_{in}(t))][(\eta_{in})\,S_\omega'(t)]^\beta[S_\omega(t)_{best}]^\gamma} & \text{, Otherwise} \end{cases} \quad (1)$$

Where 'i' represents the task and 'j' represents the VM and $S_\omega(t)_{best}$ is the best profit provider who satisfies the SLA objectives. $S_\omega'(t)$ is the optimal load balancing factor which balances the application execution using η_{ij}. $allowed_\omega(t)$ denotes the set of feasible solutions in terms of VMs. α, β, and γ parameters represent the weight of each term in PV_{ij}, are used to control the relative importance of each factor. The ACO technique applies this equation for all tasks in all VMs and it selects best VM with maximum PV_{ij} for each task using the equation (2).

$$VM_{best}=\begin{cases} \arg\max_{n\epsilon allowed_\omega(t)} [\alpha(\tau_{in}(t))][(\eta_{in})\,S_\omega'(t)]^\beta [S_\omega(t)_{best}]^\gamma & , \forall R_j = C_{ij} \\ 0 & , \text{Otherwise} \end{cases} \quad (2)$$

Each task has various PV_{ij} value while mapping with different resources. In eq.(1), if a task (T_i) is the initial task (T_1) of an application and the server, it selects the resource based on only the weight of estimated completion time and profit. Then, it constructs the remaining schedulable task list and selects the optimal resource for each task besides considering load balancing of VM resource. This process facilitates the system to execute and complete the consecutive tasks of an application. For example, there are four faces in an image and if the first two phases are scheduled in VM with high processing speed, but others in VM of less processing speed, this leads to delay the application completion time due to the lack of load balancing among tasks in the application.

The MUTUAL-BENEFIT exploits the non-recursive Dynamic programming in ACO algorithm to reduce the computation complexity. The proposed technique monitors the remote process and retains the information for scheduling decision in future. If the system receives the recursive application, it selects the processes required to execute the recursive application of the previous remote process of the same application. Moreover, it merely depends on the previous storage tasks, but not on its results. Because the dynamic programming based previous storage result is not suitable for all recursion of tasks due to a different configuration of similar applications from multiple users. For instance, image processing application includes a various process such as preprocessing, filtering, and recognition. The MUTUAL-BEEFIT takes only the process, not an exact result of previous tasks while determining similar kinds of tasks. It reduces the time complexity of MUTUAL-BENEFIT.

Even though the non-recursive dynamic programming ACO technique is sufficient to satisfy the SLA objectives and the profit of the provider, it is necessary to decide the optimal value of $S_\omega(t)_{best}$ to maintain the best trade-off between SLA objectives satisfaction and profit of the provider in the optimal VM selection.

4.3. Enhancing Profit for Cloud Provider via Optimal Resource Allocation

SLA objectives and provider's profit are inter-related with each other. If any violations occur in SLA objectives, it may affect the profit of the cloud provider in terms of increasing resource cost. Bellman's theory is used to allocate the optimal VM resources based on the utilization and resource cost of each task. With the aim of maximizing the cloud service provider's profit, MUTUAL-BENEFIT approach is targeted to achieve the load balancing of an application that also provides the long-lasting device battery. The consistent application execution on remote server

Journal of Theoretical and Applied Information Technology
15th June 2017. Vol.95. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reduces the energy cost that improves the response time and profit of the system. The profit of the provider is not static and is varied dynamically according to the cloud service provisioning by the service provider.

The consideration of determining the VM resource utilization is crucial to increase the k^{th} provider's profit while mapping the tasks to the VM resources. It balances the load of the tasks to achieve the target of the overall completion time reduction of an application using Bellman's theory which is based on the dynamic programming method by focusing initial and remaining optimal solutions. In equation (3), U(i) and cost represent the utilization and resource cost respectively. The satisfaction of mobile user is based on the utility function U(i) that measures service performance while meeting user anticipation in terms of response time. Utility function measures the user's satisfaction based on the response time of each task and the utilization of resources for completing the application execution.

$$S_{\omega}(t)_k = \sum_{i=1}^{n} \sigma * U(i)\left(\sum_{j=1}^{n} \Phi_{ij}\left(RT_{ij}^{c} + RT_{ij}^{e}\right) - cost\left(\sum_{j=1}^{n}\left(\mu P_j^{con} + \nu(P_j^{e})(\Psi_j^{e})\right)\right)\right) \quad (3)$$

Where, σ_i is the arrival rate of the i^{th} request, and Φ_{ij} is the task assigned a probability of j^{th} VM resource. RT_{ij}^{c} and RT_{ij}^{e} are the service response time of the i^{th} task in which superscript 'c' and 'e' represents at the stage of communication and execution (computation). μ and ν parameter denotes the server ON (1) or OFF (0) state and allocated (1) or not (0) respectively. P_j^{con} and P_j^{e} are the constant energy consumption of server and consumed energy in the execution stage. Ψ_j^{e} is the portion of j^{th} allocated resources to i^{th} tasks.

According to Bellman's equation, it formulates the evaluation function for allocating the resource to the selected task. This resource allocation considers the initial solution regarding utilization of each task and resources cost for executing the entire application. The relationship between the service response time and service cost finds the user satisfaction level. Utility function based allocation focuses on two factors such as user's anticipations and profit of the provider. The profit of the service provider varies according to the fluctuation of response time and cost. The service provider charges high when the user receives the immediate response from the provider. Therefore, the service provider is targeting to achieve a high level of user satisfaction and also its profit. ($S_{\omega}(t)_{best}$) are the optimal profit value and the proposed profit calculation of the system are given in the equation (4).

$$S_{\omega}(t)_{best} = \arg \max_{k \in S_P} [S_{\omega}(t)_k] \quad (4)$$

The proposed algorithm of MUTUAL-BENEFIT is shown in algorithm 1. In the algorithm, T_{SS} indicates the Selected Schedulable Task, and ξ(S(t)) represents the satisfaction of objective function based best VM resource to the corresponding task. $\Delta Ʈ_i^{\omega}(t,T')$ is the incremental update of a partial solution of the i^{th} task at time T' and R_j^{A} is the allocated VM resource which satisfies both the factors of $S_{\omega}'(t)$ and $S_{\omega}(t)_{best}$. Finally, the proposed approach uses the ThinkAir architecture to execute the allocated VMs in a parallel manner. Moreover, it optimizes the better trade-off between user satisfaction and profit of the system.

Algortihm 1: MUTUAL-BENEFIT algorithm for MCC

Input: Mobile applications and cloud resources
Output: Optimal resource allocation
//Initialization
1. Initialize tasks(T_i), resources(R_j), and pheromone value of ant(k)
2. Optimal solution=0; profit=0; load=0;
3. while allowed$_{\omega}$(t)≠0 do
4. for (i=0; i<m; i++)
5. for (j=0; j<n; j++)
6. if($T_i = T_1$) then
7. Select T_s and R_j {time, profit}
8. else
9. Select T_s and R_j {time, profit, load}
10. endif

//Finding objective based solution
11. Find E_i^{C}, E_i^{P}, and E_i^{L} of all resources
12. Optimal solution= {arg min(E_i^{C}), arg max(E_i^{P}), arg min(E_i^{L})}
13. $ŋ_{ij} = \min(E_i^{C})$
14. Sort R_j and T_i based on E_i^{C}, E_i^{P}, and E_i^{L} in ascending order

//Selection of schedulable task and resource
15. if ($T_i = T_{SS}$)
16. Find $ξ(S(t)) = [\alpha(Ʈ_{ij}(t))] [(ŋ_{ij})S_{\omega}'(t)]^{\beta} [S_{\omega}(t)_{best}]^{\gamma}$
17. Schedule T_i based on optimal
18. solution

Add T_{SS} to $S_{\omega}'(t)$, $S_{\omega}(t)_{best}$, and tabu list
endif
endfor
Update $\tau_{ij}(t)$ on pheromone table
endwhile

//Calculating optimal solution

Calculate $\Delta\tau_i^{\omega}(t,T')$ using ACO
if($\xi'(S(t)) < \xi(S(t))$) then
Optimal solution=$\xi(S(t))$
Update final optimal solution to provider
endif
Allocate R_j based on $\xi(S(t))$
R_j^A satisfies $S_{\omega}'(t)$ and $S_{\omega}(t)_{best}$
Calculate profit of the system according to (3)
endfor
endfor

5. EXPERIMENTAL EVALUATION

The proposed MUTUAL-BENEFIT approach is compared with Nested two stage game-based optimization (NTGO) [15] through evaluating the metrics of device energy, response time, and profit.

5.1. Experimental Setup

The CloudSim demonstrates MUTUAL-BENEFIT approach to executing Sudoku solver. The implementation of Sudoku solver application evaluates the performance of the proposed approach in terms of device energy and provider's profit. It considers n×n Sudoku solver table with n^2 cells and Sudoku solver has several conditions while filling digits 1 to n in cells. Consider, the mobile device solves few puzzles in the n×n table and the mobile device offloads the remaining cells based on the task complication.

The simulation is conducted in various scenarios by varying the number of mobile user requests from 500 to 2500, the level of Sudoku in terms of 'n' from 3 to 25, and the filled cells from 20 to 40%. In the resource-rich cloud server is considered as heterogeneous that has different MIPS value represents processing speed. The proposed approach is taken into the account of 5 Physical Machines (PM) resources and 25 VM resources. Each CPU has the various ranges of the energy consumption that depends on the utilization, processing time and load of the resource.

5.1.1 Evaluation Metrics

Energy saving: It is defined as the percentage of energy retained by the mobile device while executing the mobile application.

Response time: It is the interval between the service initiated time of an application and service resulted time of that application by the cloud service provider.

Application completion time: It is the overall completion time of a mobile application during mobile execution, offloading, and cloud execution.

Profit: It is the percentage of attaining profit of the provider after providing the service to the end-user. The profit measurement includes response time and energy cost with the consideration of resource utilization.

5.2 Experimental Results

5.2.1 Energy Saving

Figure 2 shows the percentage of energy savings on the mobile device while varying the complexity level of the mobile application for both MUTUAL-BENEFIT and NTGO approach with 2000 MIPS of VM resource. It indicates five complexity levels of Sudoku grid levels, such as 3×3, 6×6, 9×9, 16×16, 25×25. The percentage of energy savings of both the MUTUAL-BENEFIT and NTGO approach linearly decreases while increasing the complexity level of Sudoku from level 1 to level 5. The offloading manager of ThinkAir architecture effectively conserves the device energy, since it offloads the intensive tasks according to the device constraints. NTGO approach suddenly drops energy saving of 30% when varying the complexity level from 1 to 5, but MUTUAL-BENEFIT marginally decreases by 17% of energy saving. At the level 5, MUTUAL-BENEFIT approach saves the device energy by 13% than NTGO approach, since the exploitation of dynamic programming and parallel execution of tasks of an application.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

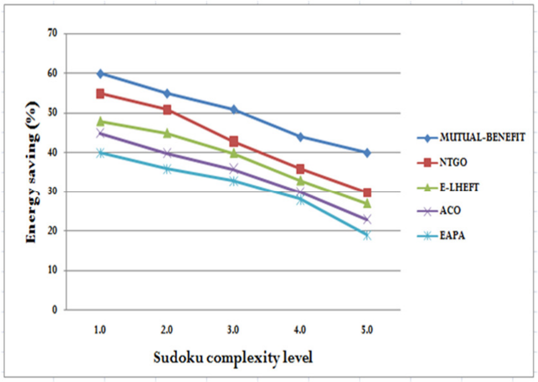

Figure 2. Energy saving (Mobile devices)

5.2.2 Response Time

Figure 3 indicates the response time of both MUTUAL-BENEFIT and NTGO approach while increasing the number of requests submitted by the mobile users and the percentage of Filled Cells (FC). The percentage of FC is referred as the ratio of a number of filling cells in the total number of cells of Sudoku solver application. The experimental evaluation of Fig.3 shows the variation of response time when FC=20% and FC=40%. The response time escalates while increasing the number of requests for the similar application.

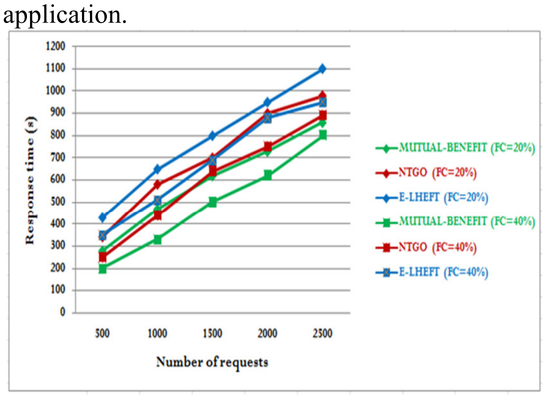

Figure 3. Response time

The ThinkAir architecture based intensive application offloading method nearly reduces the unbearable delay of the application processing. The performance of the MUTUAL-BENEFIT approach is higher than the NTGO approach after reaching 1000 number of requests, even the filled cells of the NTGO approach are higher than the MUTUAL-BENEFIT. This performance improvement is achieved by exploiting the dynamic programming assisted ACO based effective task scheduling of an application. The response time of NTGO approach increases by 9.1% while varying the number of requests from 2000 to 2500 with FC=20%. In the same scenario, MUTUAL-BENEFIT approach suddenly escalates by 10.5%, while using ACO with dynamic programming instead of using ACO with Brute-force searching method.

5.2.3 Application Completion Time

The comparative result of application completion time is shown in Figure 4 while varying the application complexity level and the percentage of FC. The overall application completion time of both MUTUAL-BENEFIT and NTGO approach slightly increases with the complexity level. The performance in terms of application completion time of NTGO approach is deviated by 24% from the MUTUAL-BENEFIT approach when the complexity level=5 and FC=40%. Application completion time depends on the satisfaction of the SLA objectives which is achieved by optimal offloading and optimal cloud execution.

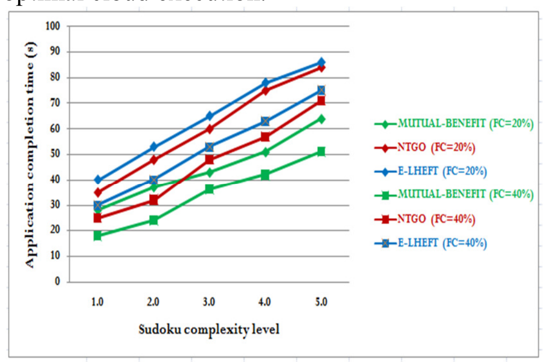

Figure 4. Application completion time

When FC=40% the performance of NTGO approach is nearly equal to the MUTUAL-BENEFIT approach when FC=20% of the points of 2 and 3 of Sudoku complexity levels, since the proposed approach shortens the longer execution time of an application using load-aware task scheduling and parallel execution.

5.2.4 Profit

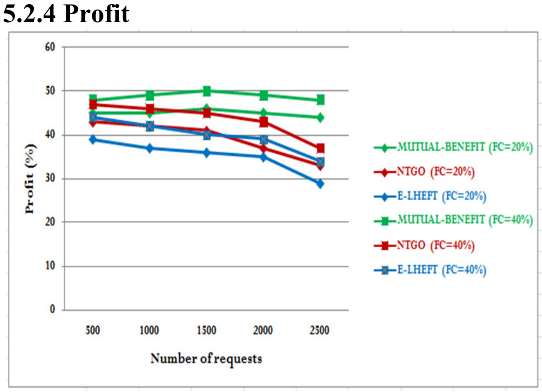

Figure 5. Profit of the provider

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 5 depicts the profit of the service provider while varying the number of requests and the percentage of filled cells. It shows a slight variation in the profit for both MUTUAL-BENEFIT and NTGO approach in which the experimental evaluation considers the provider's maximum utilization level is completed when reaching 1500 mobile user's requests. The profit of the provider slightly increases until to reach the number of requests as 1500, after that profit gets a deviation from the peak point since the profit decreases when occurring over-utilization of the resource. However, the profit of the NTGO approach continuously decreases, because it allocates the resources without the knowledge of considering the trade-off between the SLA objectives and resource cost. In MUTUAL-BENEFIT, the provider's profit assignment is corresponding to the resource cost of the particular request processing on the server. Thereby, it decreases the profit level to 11.5% more than that of MUTUAL-BENEFIT when the number of requests is 2500 and FC=40%. The profit of NTGO approach gets unexpected deviation due to the absence of trade-off consideration.

5.3 Result discussions

The MUTUAL-BENEFIT approach accomplishes the substantially minimum application completion time even increasing the complexity level of the mobile application, which is obtained by the dynamic programming and SLA objectives based task scheduling method. Although arriving the numerous mobile application requests in the cloud server, it maintains the optimal profit level by effectively utilizing the active cloud resources. The resulting, the MUTUAL-BENEFIT optimally preserves the device energy by remotely executing the mobile application in reasonable time.

The MUTUAL-BENEFIT approach significantly outperforms the existing NTGO approach in terms of device energy consumption and the profit of the service provider. The NTGO approach employs the game theory optimization method to reduce the energy consumption and maximize the profit. Even though, it lacks in providing a better performance when the complexity of the application is high and the arrival of numerous similar requests. Since, it performs the recursive process of the similar tasks and exploits the utility function merely for calculating the profit. But, the MUTUAL-BENEFIT approach focuses on the utilization of the active resources while allocating the scheduled tasks. Moreover, it utilizes the process of the recursive tasks rather than performing the repeated process in searching the optimal resources of similar applications. The MUTUAL-BENEFIT approach employs the dynamic programming method to attain the advantage of the application completion time reduction. However, it is likely to lack in providing the target performance when the number of diverse mobile applications is high than the similar application requests. Since, the diverse mobile applications have unique processing steps, which is impossible to employ the dynamic programming method. The MUTUAL-BENEFIT approach lacks in considering the time taken to restart the idle server when exploiting the idle server to the tasks, which may extend the overall completion time of application. It does not define the optimal utilization level of the resources to determine the overload and underload condition, since the 100% resource utilization may degrade the server performance.

6. CONCLUSION

This paper proposes the MUTUAL-BENEFIT approach aims to achieve the optimal task scheduling and the resource allocation to maintain the good trade-off between SLA objectives and profit of the provider. The core objective of this approach is achieved by focusing on the following phases such as Optimal offloading, task scheduling, and resource allocation. Executing dynamic programming on ACO technique, the MUTUAL-BENEFIT can contemplate the load balancing of an application. The allocation of VM resources to the corresponding tasks ensures the high profit of the provider with a higher satisfaction level of end user. The ThinkAir architecture supports to execute the tasks in parallel and to reduce the completion time of the application.

In order to satisfy the SLAs, the MUTUAL-BENEFIT approach initially exploits the ThinkAir architecture which offloads the intensive tasks to the cloud based on the energy model of the mobile device. The energy model based dynamic computation offloading prolongs the battery lifetime of the mobile device and provides the seamless application execution. Then, the MUTUAL-BENEFIT employs the dynamic programming based ACO method, which effectively schedules the intensive tasks with the consideration of SLA objectives. By

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

utilizing the dynamic programming method along with the ACO technique facilitates the execution system in reducing the additional processing time of the recursive tasks. Finally, the MUTUAL-BENEFIT approach maintains the trade-off between the SLA objectives satisfaction and profit of the provider by SLA-aware task scheduling and utility function based optimal resource allocation. The utility function focuses on the resource utilization while allocating the resources to the tasks scheduled by the dynamic programming based ACO method. The experimental result of MUTUAL-BENEFIT model significantly achieves the benefits of enhancing profit of the provider and improving performance in terms of mobile device energy and response time as compared to an existing method.

REFERENCES

[1] Saeid Abolfazli, Zohreh Sanae, Erfan Ahmed, Abdullah Gani and Rajkumar Buyya, "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges", *IEEE Communications Surveys and Tutorials,* vol.16, issue.1, pp. 337-368, 2014.

[2] Niroshinie Fernando, Seng W. Loke and Wenny Rahayu, "Mobile cloud computing: A survey", *Elsevier Transaction on Future Generation Computer Systems*, vol.29, issue.1, pp.84-106, January, 2013.

[3] Atta Ur Rehman Khan, Mazliza Othman, Sajjad Ahmad Madani and Samee Ullah Khan, "A Survey of Mobile Cloud Computing Application Models", *IEEE Communications Surveys and Tutorials*, vol.16, issue.1, pp.393-413, July, 2013.

[4] Justino Tiago and Rajkumar Buyya, "Outsourcing Resource-Intensive Tasks from Mobile Apps to Clouds: Android and Aneka Integration", *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, pp.1-8, October, 2014.

[5] Shakkeera.L and Tamilselvan.L, "QoS and load balancing aware task scheduling framework for mobile cloud computing environment", *International Journal of Wireless and Mobile Computing*, vol. 10, no. 4, pp. 309-316, July, 2016.

[6] L.Shakkeera and Latha Tamilselvan, "Energy-Aware Application Scheduling and Consolidation in Mobile Cloud Computing with Load Balancing", *Third International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA 2015), Springer India Publications,* vol. 2, pp: 253-264, August, 2015.

[7] Lin Xue, Yanzhi Wang, Qing Xie and Massoud Pedram, "Energy and Performance-Aware Task Scheduling in Mobile cloud computing Environment", *IEEE 7th International Conference on Cloud Computing*, pp. 192-199, 2014.

[8] Jian Song, Yong Cui., Minming L.i, Jiezhong Qiu and Buya.R, "Energy-TrafficTradeoff Cooperative Offloading for Mobile Cloud Computing", *IEEE 22nd International Symposium on Quality of Service (IWQoS)*, pp.284-289, May, 2014.

[9] Weiwen Zhang, Yonggang Wen, Kyle Guan, Dan Kilper, Haiyun Luo and Dapang Oliver Wu, "Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel", *IEEE Transactions On Wireless Communications*, vol.12, no.9, pp. 4569-4581, 2013.

[10] Hiromasa Yamauchi, Koji Kurihara, Toshiya Otomo, Yuta Teranish., Takahisa Suzuki and Koichiro Yamashita, "Effective distributed parallel scheduling methodology for mobile cloud computing", *In Proceedings of the 17th Workshop on Synthesis and System Integration of Mixed Information Technologies (SASIMI'12),* pp. 516–521, 2012.

[11] Mudassar Ahmad, and Ibrahim A. Cheema, "Prognostic Load Balancing Strategy for Latency Reduction in Mobile Cloud Computing", *Middle-East Journal of Scientific Research*, vol.16, issue.6, pp. 805-813, 2013.

[12] Feller Eugen, Louis Rilling and Christine Morin, "Energy-Aware Ant Colony Based Workload Placement in Clouds", *IEEE/ACM 12th International Conference on Grid Computing*, pp. 26-33, 2011.

[13] Gu. J, Hu. J, Zhao. T and Sun. G, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment", *Journal of Computers*, vol.7, issue.1, pp. 42–52, 2012.

[14] Ratan Mishra. and Anant Jaiswal, "Ant colony optimization: A solution of load balancing in cloud', *International Journal of Web and Semantic Technology*, vol.3,

issue.2, pp. 33-50, 2012.

[15] Wang.Y, Xue Lin and Massoud Pedram, "A Nested Two Stage Game-Based Optimization Framework in Mobile Cloud Computing Cystem", *IEEE 7th International Symposium on Service Oriented System Engineering (SOSE)*, pp. 494-502, March, 2013.

[16] Pelin Angin and Bharat Bhargava, "An agent-based optimization framework for mobile-cloud computing", *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol.4, issue.2, pp. 1-17, 2013.

[17] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier and Xinwen Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in cloud for mobile code offloading", *IEEE Proceedings INFOCOM*, pp. 945-953, 2012.

[18] Kosta Sokol, Andrius Aucinas, Pan Hui, Richard Mortier and Xinwen Zhang, "Unleashing the power of mobile cloud computing using ThinkAir", *arXiv preprint arXiv:1105.3232*, 2011.

[19] Hongbin Liang., Tianyi Xing., Lin X. Cai., Dijiang Huang., Daiyuan Peng. and Yan Liu, "Adaptive computing resource allocation for Mobile cloud computing", *International Journal of Distributed Sensor Networks*, 2013.

[20] 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 (ISLPED' 12)*, pp. 279-284, 2012.

[21] Yuan Zhao., Lei Zhang, Xiaoqiang Ma, Jiangchuan Liu and Hongbo Jiang, "CAME: cloud-assisted motion estimation for mobile video compression and transmission", *ACM on Proceedings of the 22nd International workshop on Network and Operating System Support for Digital Audio and Video*, pp. 95-100, 2012.

[22] Rahimi, M. Reza, Nalini Venkatasubramanian, Sharad Mehrotra and Athanasios V. Vasilakos, "MAPCloud: Mobile applications on an Elastic and Scalable 2-Tier Cloud Architecture", *In Proceedings of IEEE Fifth International Conference on Utility and Cloud Computing (UCC)*, pp. 83-90, 2012.

[23] Boloor Keerthana., Rada Chirkova, Yannis Viniotis. and Tiia Salo, "Dynamic Request Allocation and Scheduling for Context Aware Applications Subject to a Percentile Response Rime SLA in a Distributed Cloud", *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 464-472, November, 2010.

[24] Goudarzi Hadi and Massoud Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems", *IEEE International Conference on Cloud Computing (CLOUD)*, pp. 324-331, 2011.

[25] Yang Zhang, Dusit Niyato and Ping Wang, " An Auction Mechanism for Resource Allocation in Mobile Cloud Computing systems", In Wireless Algorithms, Systems, and Applications, Springer Berlin Heidelberg, pp. 76-87, 2013.

[26] 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.