Vol. 5(16) Jul. 2015, PP. 2327-2339

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

OCSRAM: An Optimized Framework for Dynamic Service Adaptation in Mobile Cloud Computing Systems

Arash Ghorbannia Delavar[1], *Somayyeh Zahedian Pishkhani*[1*] *and Omid Mehdizadeh Tourzan*[2]

[1]*Department of Computer Science, Payame Noor University, PO BOX 19395-3697, Tehran, Iran*

[2]*Department of Computer Science, Iran University of Science and Technology, Tehran,* Iran

**Corresponding Author's E-mail:* s.zahedian@gmail.com

Abstract

Restrictions such as heterogeneity of cloud resources and processing power of mobile phones provide mobile users with an optimized service which is considered as an interesting and controversial challenge based on the Service Level Agreement (SLA) in mobile cloud computing environment. In this article, a modular framework named Optimized CSRAM (OCSRAM) is proposed as an expansion for previous proposed framework (CSRAM) which tries to keep the real-time nature of the system by computation offloading as well as better usage of evolutionary nature of Genetic Algorithm (GA) in adaptation process. In addition, not only does it use more effective environmental parameters as inputs, but, in order to increase its flexibility and reliability compared with CSRAM, it also considers failure recovery process in modular design of the system.

Keywords: *Mobile Cloud Computing, Service Discovery, Context Awareness, Quality of Service, Service Request Adaptation, Failure Recovery.*

1. Introduction

With increasing use of power constrained devices like smartphones, PDAs, etc. and also availability of fast and reliable wireless networks, people are more interested in using mobile applications and cloud computing platforms to have their desired services. Normally, with the change of mobile devices' local context (e.g. resources, environmental variables and user preferences) during service runtime, the initial requested quality of service (QoS) level by mobile application may not match the capabilities of the device in its current context. Therefore, the considered mismatch not only results in resource wastage (e.g. overusing phone resources including battery life and processing power) and higher monetary cost, but also it leads up to user dissatisfaction with the quality of received service [1], [2], [3]. As mobile cloud computing represents the technology of mobile services in cloud environments, providing mobile users with an optimized quality of service based on the changing user's context information is one of the important issues in mobile cloud systems [3]. Furthermore, the experimental comparisons and performance evaluations demonstrate that using different contextual information such as hardware resources, environmental variables, user preferences, etc. as input parameters in request adaptation process elevates the degree of accuracy and real precision of service discovery mechanism. On the other hand, ensuring the failure recovery process is essential in real-time distributed systems, because it helps to meet end-to-end system requirements. In fact, the recovery process must be achieved in a timely fashion so that response time of clients is maintained [4]. With regards to aforementioned issues, one possibly

Article History:

Received Date: Mar. 02, 2015
Accepted Date: Jun. 26, 2015
Available Online: Jul. 09, 2015

promising solution seems to be using a recovery-oriented system for monitoring phone resources and context information while dynamically adapting cloud services to it. This adaptation is supposed to happen gradually so that the best allocation of resources takes place all time [5]. Moreover, in this middleware, failure recovery process can be done in different modules levels to make the whole system as reliable as possible [4].

In CSRAM framework [5], authors are trying to dynamically adapt mobile users' request in cloud platform by keeping the real-time nature of the system, and simultaneously, increasing the speed and accuracy of response by offloading some part of computations to the cloud. In addition, in CSRAM framework [5], the service provider's context information as well as user's information are used to increase the precision of service selection process for VIP users. Due to the copious number of environmental parameters, this issue is considered as an NP-hard problem. Therefore, Genetic algorithm [6] is used in the request adaptation process of CSRAM framework. Using Genetic Algorithm not only helps make the adaptation mechanism more flexible in real-time change of problem searching area but also make the adaptation function independent of the type of the requested service.

In this article, a framework called OCSRAM is proposed which is an optimized version of CSRAM framework. In this newly proposed framework, the role of failure recovery process of different modules is considered to increase the reliability of the whole system. In addition, a customized Genetic Algorithm is used in request adaptation process as an attempt to decrease the service response time. Moreover, other environmental parameters are added to the input list of the problem to make the result of service discovery more accurate. Finally, the system modules are broken down into a more detailed and transparent schema.

The organization of this article is as follows. In section 2, the related works are discussed. Then in section 3, the proposed framework is introduced. Simulation results of OCSRAM are then presented in section 4, and eventually the paper is concluded in section 5.

2. Related Works

In Recent years, web services adaptation by user's context information, alongside providing optimized quality of service has been of great importance to researchers and consequently, various frameworks and algorithms have been proposed in this regard.

Frameworks like CARISMA [7], CHISEL [8], MUSIC [9] and PLASTIC [10] try to adapt the service request on variable context information and predefined policies basis. In Q_CAD [11] optimum service regarding the context and QoS information is selected. However, it does not support the dynamic adaptation process of service request at runtime.

DINO [12] uses a broker-oriented mechanism for adaptation process. Adaptation mechanism proposed by Song et al. [13] is done based on a utility model definition and use of the users' and the service providers' information. Soukkarieh et al. [14] presented architecture in which firstly, a service is delivered, and then at the runtime, the dynamic adaptation process based on the comparison of the user's context with the context of the primarily delivered service is carried on. Li et al. [15] proposed a framework which utilizes HAC algorithm as the adaptation method for proactive service discovery in pervasive environments. The adaptation mechanism offered by SMICLOUD [16] utilizes AHP algorithm for rating available cloud services.

Jung et al. [17] presented a conceptual framework for providing context-aware services through detection of any type change cause and accordingly, choosing a suitable adaptor. However, predefined list of causes and adaptors leads to inflexibility of this approach in real world. Peng et al. [18] introduced a QoS-aware system that monitors and assesses the QoS parameters and accordingly performs adaptation to (with) an appropriate QoS mode. In another work, Qing et al. [19] adapted the quality of service using FCM adaptation algorithm. The adaptation process is performed by analyzing user's context information and comparing different available priorities. The considered priorities are obtained by weighing different service parameters and user information.

VOLARE proposed by Panayiotis et al. [20] is a middleware module embedded in mobile devices that monitors resources and contexts of them, and dynamically adjusts user requirements at runtime. Focusing on the total adaptation process on the mobile phones with limited storage and processing resources decreases their efficiency in real-time applications [21].

Go et al. [22] presented a preference-aware skyline service discovery system in the mobile cloud system. Their proposed system is composed of zone-based overlay network in which the service request is generated with user preference extracted by a Singular Value Decomposition method from the user's service access history. DaaS Proposed by Elgazzar et al. [23] is a cloud-based discovery framework that utilizes web service clustering for ranking relevant services regard to user context and preferences information.

Tahir et al. [24] proposed a model for service management in mobile cloud infrastructures. This research presents a decentralized model that gets help from the flocking behavior of birds for configuring mobile cloud infrastructure the aim of this system is providing different services based on social relations in a decentralized manner.

2.1. CSRAM Framework

CSRAM framework is a middleware solution which evaluates the received service requests and then adapts it according to the context information of user and service provider as well as the QoS information. Then, it sends the adapted request to the cloud.

In CSRAM framework, the service discovery phase begins right after a user service request is received. In this phase, Request Management Module (RMM) receives the user context information from Context Management Module (CMM) and uses it for analyzing and processing the received request from application. Then, it transmits the processed request to Service Management Module (SMM) which is on the cloud. If the user requesting a service is a VIP user, SMM asks the service providers to send the context information of their services. Then, it submits the integrated service request to Adaptation Management Module (AMM) and receives the adapted request from this module. It runs the best service selection process regarding to the provided data afterwards. Thus AMM sends the adapted request to the considered service provider. Finally, it binds the service to the user. Figure 1 represents the flowchart of service request management process in CSRAM framework.

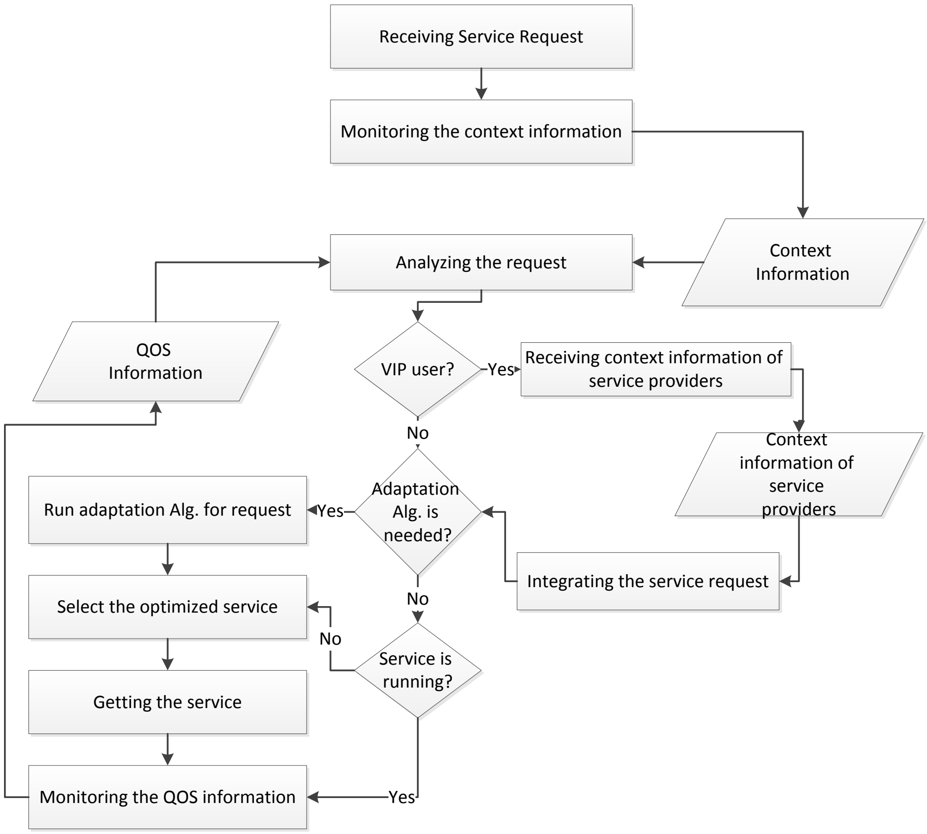

Fig. 1. Flowchart of service request management process in CSRAM [5]

Soon after the service discovery phase is completed, the runtime phase begins. In runtime phase the context and QoS information is monitored steadily and updated requests is sent to the cloud. At the same time, the context information of the service provider is monitored and sent to the cloud. AMM collects the received data, and if there is a difference between available quality level and the expected level in the service, it retransmits the adapted request for selecting the most appropriate service.

The purpose of CSRAM is keeping the nature of real-time systems as well as increasing speed and accuracy of the provided response, by offloading computational part of request adaptation process to the cloud. As another considerable advantage of this approach, context information of the service providers is efficiently utilized to provide VIP users with a careful selection of an optimized service.

In the newly proposed framework, it is tried to perform an optimization on CSRAM framework. Therefore, failure recovery process is embedded in OCSRAM in order to increase its reliability. In addition, using a customized GA in the request adaptation process helps decrease the service's response time. Moreover, obtained results indicate that considering more environmental parameters such as user preferred usage cost and available processor and storage of service provider in the input list of the problem makes the result of service discovery more accurate. Finally, breaking down the system modules helps represent them in a more detailed and transparent schema.

3. Optimized Framework: OCSRAM

As previously stated, the main objective of OCSRAM is to maintain the original structure of CSRAM framework, and to improve system functionalities at the same time. One approach to fulfill the aforementioned objective is inserting failure recovery module to CSRAM framework in order to increase system reliability. The purpose of failure recovery module is to determine procedures and standards of recovery plans taking place at the time of system's module or component failures. The described procedures provide a structured recovery plan used for recovery management whenever a failure is diagnosed.

Another difference is customization of request adaptation process carried on by AMM which is achieved by applying historical information in decision making by prioritizing previous optimized solutions. Comparing with CSRAM which uses genetic algorithm as the whole adaptation mechanism, results prove that applying such an adaptation method leads to less response time and lower computational complexity. In the meantime, expanding the input list of the problem increases the accuracy of adaptation process. Finally, breaking down the modules of OCSRAM framework, helps represent the system in a more detailed schema.

Figure 2 shows the newly optimized framework of Service Request Management based on context and QoS information. In addition, Table 1 summarizes the functions of various modules and their associated components in the framework.

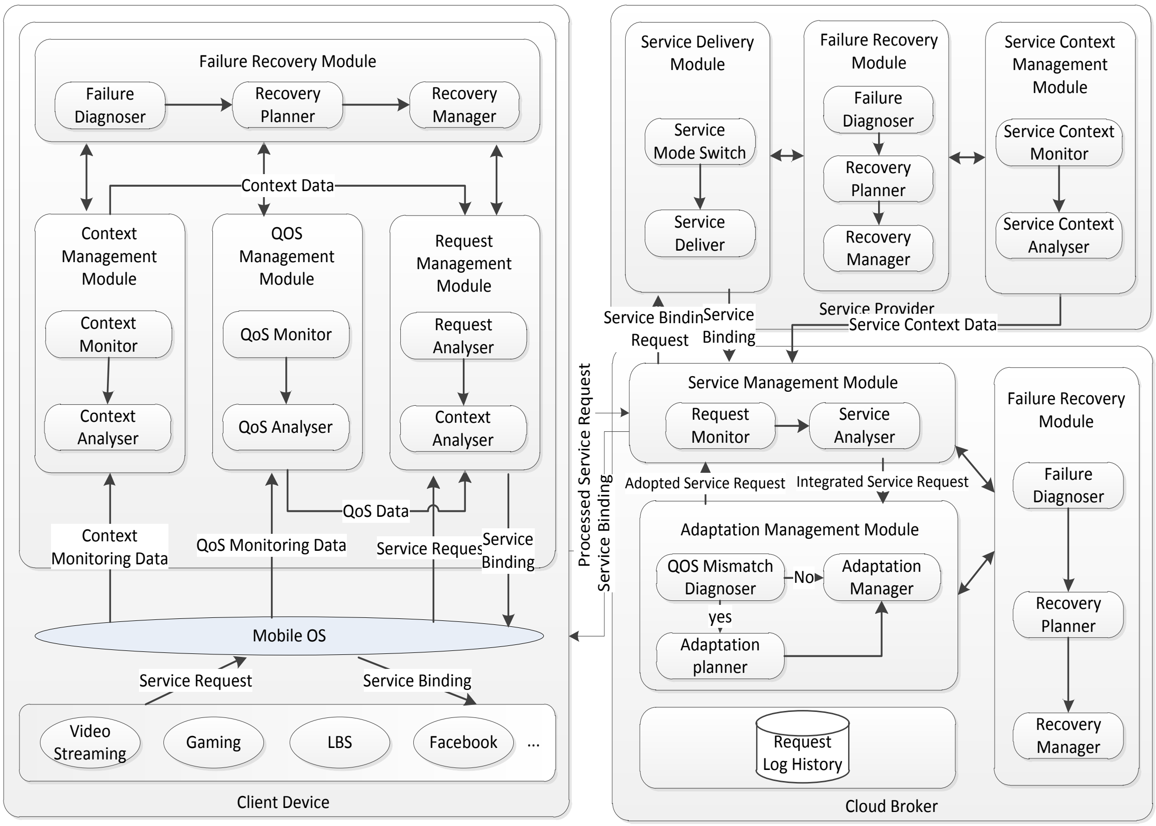

Fig. 2. Service request management of OCSRAM

3.1. Proposed Process

The overall process of user request management and adaptation in OCSRAM is similar to CSRAM framework. In service discovery phase, the request received from the application is processed based on user context data and service context data in case of VIP user. Then, adaptation process is performed on the restructured service request and the adapted request is sent to the selected service provider. Runtime phase begins right after directing the appropriate service to the user. In this phase, context and QoS data is monitored continually and if there is a

Table 1. Functionality of OCSRAM different modules

Module	Related components	Functionality
Request management Module	• Request Analyzer • Service Binder	• Creating/reconstructing the body of the request considering the context and QoS information and sending it to the broker • Navigating the binded service to the user
Context Management Module	• Context Monitor • Context Analyzer	• Monitoring the different qualitative or quantitative context data • Analyzing the context data and converting it to the required information format
QoS Management Module	• QoS Monitor • QoS Analyzer	• Monitoring the different qualitative or quantitative QoS data • Analyzing the QoS data and converting it to the required information format
Service Management Module	• Request Monitor • Service Analyzer	• Getting the body of service request and in case of a VIP user, the context information of the service provider • Integrating the request based on the considered format and send it for adaptation • Selecting the best service mode based on received adapted request and navigating the received service mode to the user
Service Context Management Module	• Service Context Monitor • Service Context Analyzer	• monitoring the different qualitative or quantitative service context data • analyzing the service context data and converting it to the required information format
Adaptation Management Module	• Mismatch Diagnoser • Adaptation Planner • Adaptation Manager	• Diagnosing the changes in user's status via processing the received request • Planning and managing appropriate adaptation method if needed
Service Delivery Module	• Service Mode Switch • Service Deliver	• providing the user with the appropriate service mode regarding to the received request
Failure Recovery Module	• Failure Diagnoser • Recovery Planner • Recovery Manager	• Diagnosing any failure accrued in module or component level • Planning and managing the failure recovery process

mismatch between service and available quality level, adaptation process is performed and the adapted request is sent to the cloud.

The adaptation process is slightly different in OCSRAM framework. In the new scenario, a combination of GA and LOGs stored in the system is being used in order to find the optimal solution in the least possible time. In each iteration, the values of input parameters as well as the selected solution is recorded in the format of the system logs. At discovery time, GA is used to search the problem area and to reach the optimal adaptation of service requests.

At runtime, AMM first checks any changes in the user's context data compared to the last recorded information; for VIP users, the service provider's data is also included in the checking process. Purpose of this checking is to decide whether the service request needs any adaptations or not. This is performed by fitness computation of received request and comparing it with the corresponding fitness value of the last optimized request. If their difference is less than a predefined threshold, the current service mode can be considered as an appropriate mode for user. Otherwise, adaptation process is needed by which the considered request is compared with all previous optimized requests recorded in the system in order to find the best solution regarding to the considered mismatch threshold. If such an optimized request is found, the applied solution corresponding to current situation will be chosen as optimal solution. Otherwise, the GA will be chosen to search the problem area and find the best state of adaptation request. Thus, by trusting prior decision the optimum service is selected at the least amount of time. This approach is promising in different situations especially when a significant difference is not observed in values of input parameters in sequential iterations. Table 2 represents the pseudo code of the proposed algorithm.

Table 2. Pseudo Code of Adaptation Module in OCSRAM Framework

Input: Req_i := i^{th} integrated request; β:= request mismatch threshold

Output: $Solution_i$ as optimized request

```
If  i:=1 then
  optimizedReq_i := genetic_Algorithm();
Else
   If mismatch (Req_i , optimizedReq_i-1) < β
     optimizedReq_i := optimizedReq_i-1;
   Else
     optimizedReq_i := min_Mismatch();
       If optimizedReq_i is null
          optimizedReq_i := genetic_Algorithm();;
          End if
       End if
     End if
     Return  optimizedReq_i as optimized solution
```

Figure 3 and 4 show the Business Process Model and Notation (BPMN) of OCSRAM service request management and adaptation process, respectively.

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)
Universal Scientific Organization, www.aeuso.org
PISSN: 2411-6173, EISSN: 2305-0543

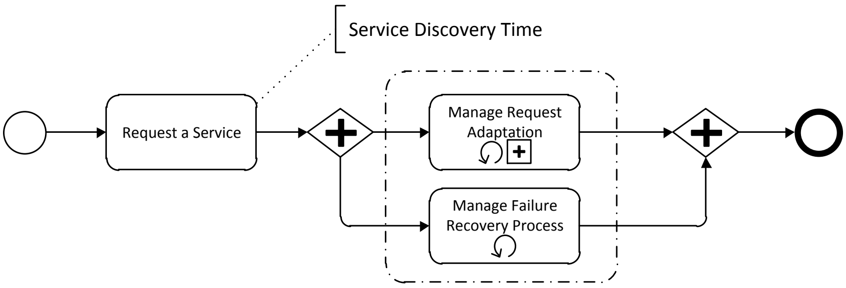

Fig. 3. BPMN diagram of OCSRAM system

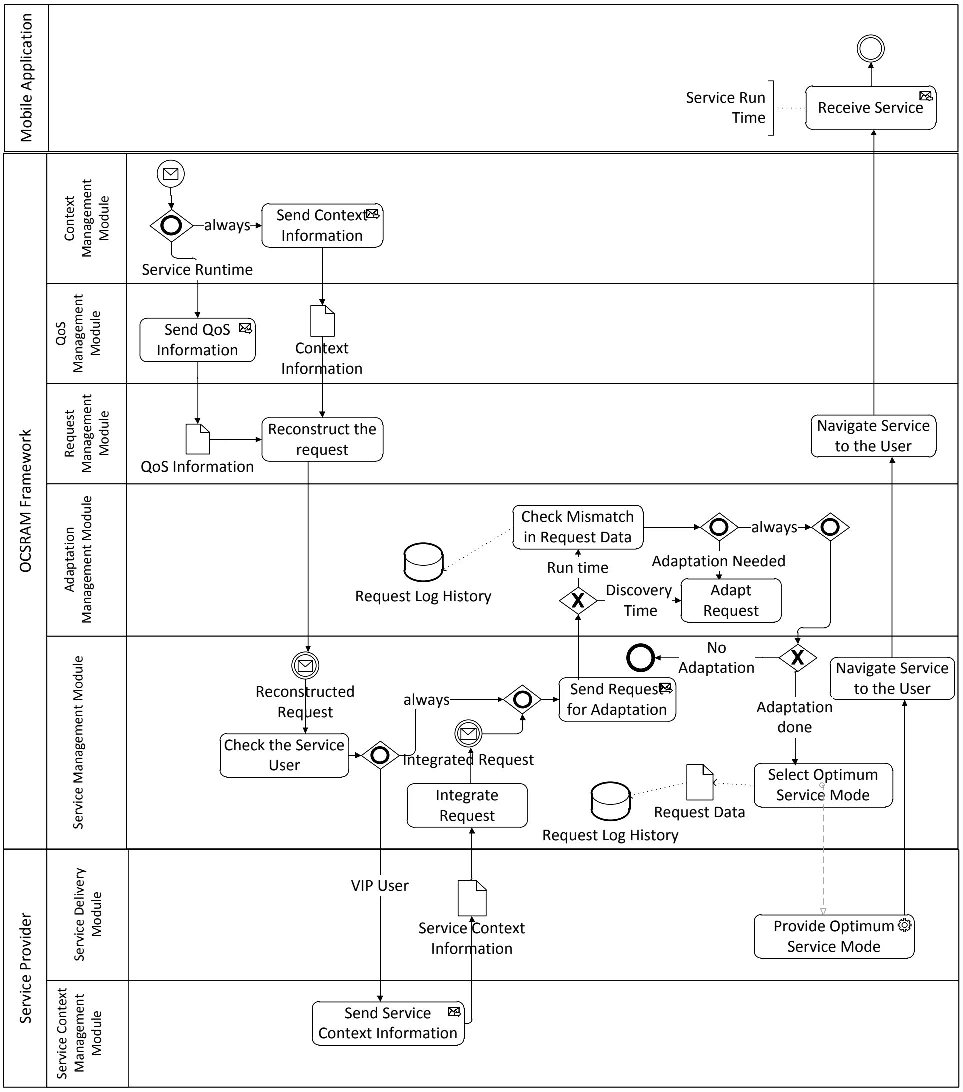

Fig 4. BPMN Diagram of OCSRAM Adaptation Process

4. Experimental Results

4.1. Case Study

In this section, CSRAM's case study is considered as a benchmark for evaluating the results of OCSRAM framework and comparing them with those of CSRAM. As before, the user can subscribe to one of the four service modes of video streaming service (i.e. 28, 56, 128 or 256 kbps) according to the SLA. However, because of different network limitations such as bandwidth drop , the user cannot use the desired choice. By default, the user should pay the usage cost indicated in the SLA. In fact, the purpose of both CSRAM and OCSRAM frameworks is reduction of usage cost and converting it to the real usage cost. In this section, how effective the new optimized framework is to reduce the cost is evaluated. For the sake of simplicity, the video decoding process on mobile phone is ignored and it is assumed that the user receives streaming packets.

4.2. EVALUATION PARAMETERS

As previously mentioned, it is aimed not only to make the selected solution close to the reality but also to have a more accurate adapted request based on both user's and service provider's contextual information in OCSRAM. This task is done by considering more service and mobile phone context, and also QoS parameters in adaptation process. As before, there are different chromosomes in each generation with the same context, but different QoS values. Those chromosomes are shown by cc_i and qs_j, respectively. Furthermore, sc_k represents the context information of service provider. The equations (1), (2), (3) and (4) show considered parameters and related weights, and equation (5) shows the utilized fitness function. Consequently, eleven parameters are considered for determining solution's optimality.

$$cc_i \in \begin{bmatrix} \text{Storage, Battery, CPU,} \\ \text{Bandwidth, preferedCost} \end{bmatrix}, i \in [1..5] \qquad (1)$$

$$qs_j \in \begin{bmatrix} \text{Responsebandwidth, Availability,} \\ \text{Response Time, Service Cost} \end{bmatrix}, j \in [1..4] \qquad (2)$$

$$sc_k \in \begin{bmatrix} \text{AvailableCPU,} \\ \text{AvailableStorage} \end{bmatrix}, k \in [1..2] \qquad (3)$$

$$\sum_{i=1}^{5} W_i + \sum_{j=1}^{4} W_j + \sum_{k=1}^{2} W_k = 1 \qquad (4)$$

$$\text{Fitness}_{\text{total}} = \sum_{i=1}^{5} W_i cc_i + \sum_{j=1}^{4} W_j qs_j + \sum_{k=1}^{2} W_k\, sc_k$$

$$\left(W_{i=5} = \frac{1}{W_i}, W_{j=3,4} = \frac{1}{W_j}\right) \qquad (5)$$

4.3. EVALUATION RESULTS

CSRAM and OCSRAM frameworks are both performed on defined data set and their outputs are compared. The execution parameters of the new optimized system are as follows:

- Request mismatch threshold: 10
- Wi∈[0.05,0.05,0.05,0.2,0.05] , i ∈[1..5]
- Wj∈[0.1 ,0.1,0.25,0.05] , j ∈[1..4]
- Wk∈[0.05,0.05] ,k∈[1..2]
- Number of executions at different time instances: 20
- The iteration number of genetic algorithm: until obtaining a solution
- The iteration number of adaptation process: 24
- The number of population members at any given moment: 256
- The rate of Crossover: 0.005
- The rate of Mutation: 0.005
- The selection size: 0.75

Figure 5, 6 represent the service response time and availability of both approaches, respectively. Results indicate 50% reduction in response time and %0.1 increase in video streaming service availability in OCSRAM. Regarding the structural changes, adaptation process improvement and ultimately increasing the selected solution's accuracy in OCSRAM framework are expected.

In addition, Figure 7 considers both battery and bandwidth as instances of problem input parameters and represents the amount of money saved for both approaches . According to Figure 7, the saved amount is %27 which is %3 worse than that of CSRAM approach with %30 saving in user costs. This service cost degradation can be justified considering other improvement parameters such as response time, availability and accuracy of provided service.

Table 3 illustrates the detailed evaluation results regarding different parameters. In addition, Table 4 shows the number of considered parameters and the related results for each one.

Results prove that OCSRAM framework has successfully applied the desired

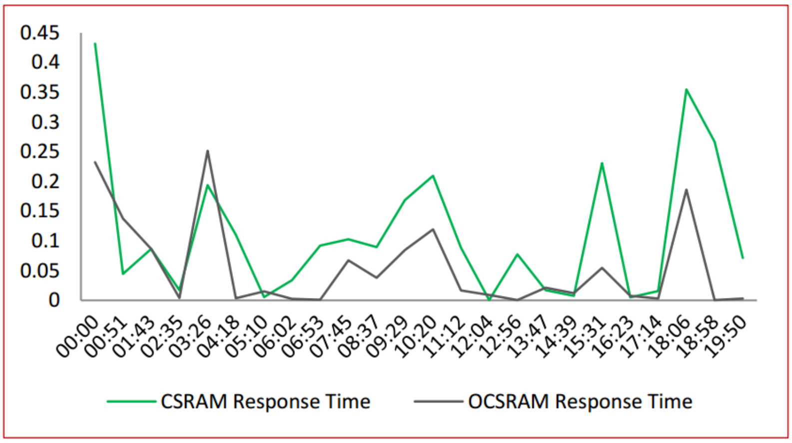

Fig 5. Service response time compared with CSRAM

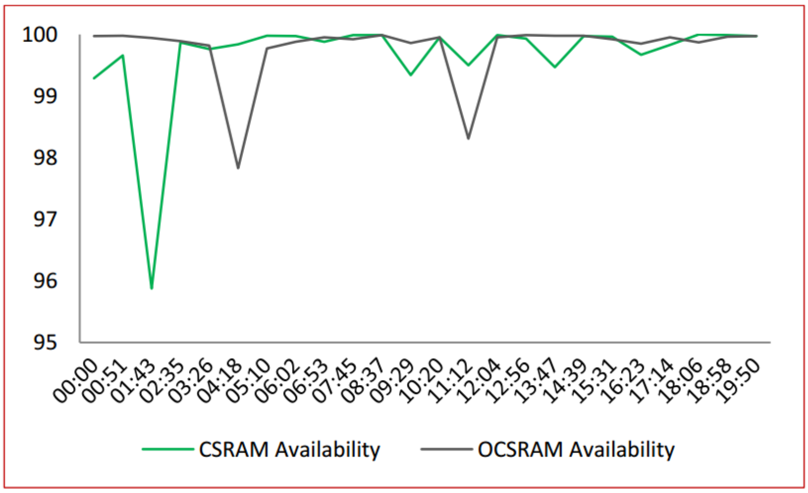

Fig 6. Service availability compared with CSRAM

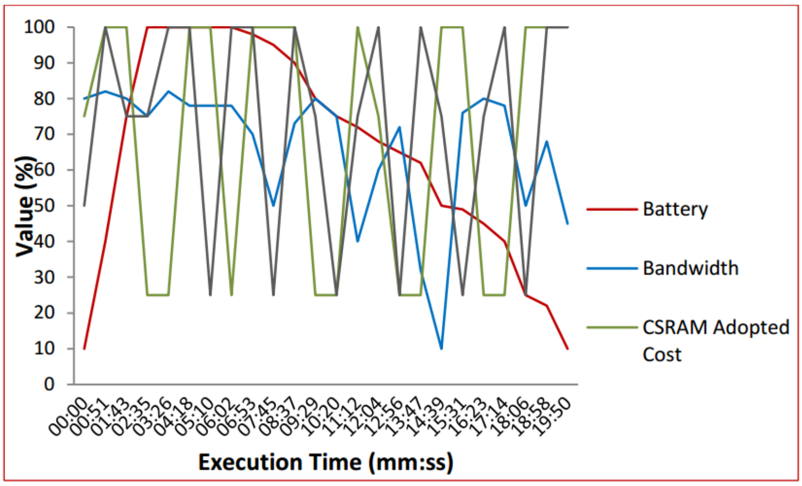

Fig 7. Percentage of cost saving compared with CSRAM

Table 3. The values of involved parameters in OCSRAM

Objectives / Time (mm:ss)	SC_Storage (%)	SC_Battery (%)	SC_CPU (%)	SP_CPU (%)	SP_Storage (%)	SC_Bandwidth (%)	SC_preferedCost (%)	Res bandwidth (kbps)	Availability(%)	Response Time (s)	Service Cost (s)
0:00	43	10	10	45	56	80	50	43	99.97	0.2325	0.2421
0:51	56	40	22	68	34	82	50	223	99.98	0.1379	0.3661
1:43	60	75	25	50	35	80	100	119	99.94	0.0866	0.2054
2:35	35	100	40	78	65	75	100	89	99.89	0.0042	0.2090
3:26	56	100	45	80	67	82	100	205	99.82	0.2515	0.3123
4:18	70	100	50	10	32	78	80	189	97.83	0.0036	0.5258
5:10	64	100	62	32	45	78	80	17	99.77	0.0155	0.2220
6:02	47	100	65	72	72	78	80	248	99.88	0.0025	0.2410
6:53	23	98	68	60	57	70	100	165	99.95	0.0011	0.6501
7:45	17	95	72	40	26	50	80	21	99.92	0.0673	0.8112
8:37	32	90	75	75	59	73	100	170	99.99	0.0382	0.3491
9:29	47	80	80	80	72	80	100	101	99.86	0.0843	0.2620
10:20	72	75	90	73	60	75	100	15	99.95	0.1194	0.2824
11:12	56	72	95	50	42	40	50	98	98.31	0.0166	0.3925
12:04	68	68	98	70	47	60	50	168	99.95	0.0093	0.2057
12:56	83	65	100	78	57	72	50	17	99.99	0.0006	0.2559
13:47	48	62	100	78	56	32	100	212	99.98	0.0214	0.2010
14:39	45	50	100	78	87	10	100	112	99.98	0.0124	0.2951
15:31	27	49	100	82	69	76	50	12	99.92	0.0549	0.2317
16:23	64	45	100	75	66	80	100	104	99.85	0.0077	0.2083
17:14	79	40	75	80	69	78	100	167	99.95	0.0031	0.7054
18:06	93	25	40	82	49	50	50	18	99.87	0.1862	0.2631
18:58	83	22	10	80	32	68	50	207	99.96	0.0006	0.8827
19:50	65	10	25	68	50	45	50	213	99.97	0.0030	0.2042

Table 4. CSRAM framework vs. OCSRAM framework

	Involved Parameters	Percentage of Saving	Adaptation Model
CSRAM Framework	7	30%	Genetic Alg.
OCSRAM Framework	11	27%	Heuristic Adaptation Alg.

changes on CSRAM framework. In addition, the related adaptation process of OCSRAM, involves the real-time nature of mobile cloud services in choosing the final solution.

CONCLUSIONS

In this article, an effective framework called OCSRAM is proposed. OCSRAM is derived from applying considerable modification on CSRAM. In the previous work, Comparing CSRAM with similar common frameworks showed that CSRAM framework decreases the amount of imposed calculations to user's mobile phone. CSRAM considers various environmental parameters for obtaining results with high precision. In addition, its flexibility makes it possible to use other optimization algorithms rather than genetic algorithm to change the evaluation criteria. In this paper, by applying a heuristic adaptation algorithm as the search method, it is attempted to optimize search algorithm to decrease the calculation cost and to speed up searching process. In this method, a log history from previous optimized solutions is used. Genetic Algorithm (GA) is used for finding the optimal solution at discovery phase. At runtime, the recorded log is reviewed first and then, if the obtained solutions are close enough to the logged solutions, search process stops and the best solution is selected as optimal solution. Otherwise, the optimized solution is found by GA. Comparing with CSRAM, more effective environmental parameters are considered as problem inputs to make solutions more accurate in OCSRAM framework. Finally, embedding failure recovery process in the modular design of OCSRAM framework increases its flexibility and reliability.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the Clouds: A View of Cloud Computing", Commun of the ACM, (2010).

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," Future Generation Computer Systems, vol. 29, no. 1, pp. 84–106 (2013).

[3] 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).

[4] F. Wolf, J. Balasubramanian, S. Tambe, A. Gokhale, D. C. Schmidt, "Supporting component-based failover units in middleware for distributed real-time and embedded systems," Journal of Systems Architecture, vol. 57, no. 6, pp. 597-613 (2011).

[5] S. Zahedian, A. Gh. Delavar, Y. Aryan, "A framework for optimizing the response quality of services in mobile cloud computing systems," IJCA, vol. 89, no. 17, pp. 27-34 (2014).

[6] J.M. Renders, S.P. Flasse, "Hybrid methods using genetic algorithms for global optimization," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 26, No. 2, pp. 243-258 (1996).

[7] L. Capra, W. Emmerich, C. Mascolo, "CARISMA: Context-Aware Reflective Middleware System for Mobile Applications, IEEE Transactions on Software Engineering, Vol. 29, No. 10, pp. 921-945 (2003).

[8] J. Keeney and V. Cahill, "Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework," Proc. Fourth IEEE Int'l Workshop Policies for Distributed Systems and Networks, pp. 3-14 (2003).

[9] R. Rouvoy, P. Barone, Y. Ding, F. Eliassen, S. Hallsteinsen, J. Lorenzo, A. Mamelli, U. Scholz, "MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments," LNCS 5525, Software Engineering for Self-Adaptive Systems, pp. 164-182 (2009).

[10] Bertolino, W. Emmerich, P. Inverardi, V. Issarny, F. Liotopoulos, P. Plaza, "PLASTIC: Providing Lightweight Adaptable Service Technology for Pervasive Information & Communication," Automated Software Engineering (2009).

[11] L. Capra, S. Zachariadis, C. Mascolo, "Q-CAD: QoS and context aware discovery framework for adaptive mobile systems," Int. Conf. Pervasive Services (ICPS), pp. 453–456 (2005).

[12] A. Mukhija, A. Dingwall-Smith, D. Rosenblum, "Qos-aware service composition in dino," ECOWS, pp. 3-12 (2007).

[13] H. Song, C. S. Bae, J. W. Lee, C.-H. Youn, "Utility adaptive service brokering mechanism for personal cloud service," MILCOM, pp. 1622-1627 (2011).

[14] B. Soukkarieh, F. Sèdes, "Dynamic Services Adaptation to the User's Context," Int. Conf. on Internet and Web Applications and Services, (2009).

[15] F. Li, K. Rasch, H. Truong, R. Ayani, S. Dustdar, "Proactive Service Discovery in Pervasive Environments", ICPS, pp. 126-133 (2010).

[16] S.K. Garg, S. Versteeg, R. Buyya, "SMICloud: a Framework for Ranking of Cloud Computing Services," Future Generation Computer Systems, Vol. 29, No. 4, pp. 1012-1023 (2013).

[17] H. H. La, S. D. Kim, "A Conceptual Framework for Provisioning Context-aware Mobile Cloud Services," IEEE Int. Conf. Cloud Computing (CLOUD), pp. 466-473 (2010).

[18] P. Zhang and Z. Yan, "A QoS-AWARE SYSTEM FOR MOBILE CLOUD COMPUTING," IEEE CCIS, pp. 518-522 (2011).

[19] W. Qing, L. Zhenbang, Y. Yuyu, Z. Hong, "Adaptive Service Selection Method in Mobile Cloud Computing," China Communications, Vol. 9, No. 12, pp. 46–55 (2012).

[20] P. Papakos, L. Capra, D. S. Rosenblum, "VOLARE: Context-Aware Adaptive Cloud Service Discovery for Mobile Systems," Intl. Workshop on Adaptive and reflective Middleware (ARM), pp. 32-38 (2010).

[21] Ch. Shan, A. Gawde, "Service based offloading from mobile devices into the cloud," Int. J. Advanced Computational Engineering and Networking, Vol. 1, No. 9 (2013).
[22] Y. Go, H. Song, "Preference-aware skyline service discovery in the mobile cloud system," Personal Indoor and Mobile Radio Communications (PIMRC), pp. 3543 - 3547 (2013).
[23] Kh. Elgazzar, H. S. Hossam, P. Martin, "DaaS: Cloud-based mobile Web service discovery," Pervasive and Mobile Computing, Vol. 13, pp. 67-84 (2013).
[24] A. O. Tahir, A. Adeyinka, "Autonomic Service Management in Mobile Cloud Infrastructures," Int. J. New Computer Architectures and their Applications (IJNCAA), Vol. 4, No. 2, pp. 91-107 (2014).

Authors

Arash Ghorbannia Delavar received the MSc and Ph.D.degrees in computer engineering from Science and Research University, Tehran, IRAN, in 2002 and 2007. He is currently an assistant professor in the Department of Computer Science, Payame Noor University, Tehran, IRAN. His research interests are in the areas of computer networks, data mining.

Somayyeh Zahedian Pishkhani received the BS, in computer engineering from Iran University of Science and Technology in 2005 and now, she is a Student of MS degree in the department of Computer Engineering and Information Technology in Payame Noor University, Tehran, IRAN. Her research interests include cloud computing, distributed systems and value added services.

Omid Mehdizadeh Tourzan received the BS, in computer engineering from Iran University of Science and Technology in 2008 and now, he is working as a network engineer at Ericsson Company, Tehran, IRAN. His research interests include cloud computing, telecommunication and artificial intelligence.