Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 3 Özel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 3 Special Issue (2015)
ISSN: 1300-1949

A new method for re-allocation resource for Real-Time Service Oriented Architecture

Marzieh MASOUDİ[1], Sima EMADİ[2,*]

[1]*Department of Computer Engineering, Maybod Branch, Islamic Azad University, Yazd, IRAN*

[2]*Department of Computer Engineering, Yazd Branch, Islamic Azad University, Yazd, IRAN*

Received: 01.02.2015; Accepted: 05.05.2015

Abstract. In recent years, Real-Time service oriented architecture has been highly considered. Such architecture is used in many business processes for loose connection services integration that is being utilized more by passing time. One of significant challenges of transferring service oriented aspects to Real-Time is to provide guarantee any kind of service performance before implementation time. In a way to guarantee its required resources for any service within requested time period. In this research, it recommended to a method for more utilization of resource free time and re-allocation of such time for services which have not gained all their required time.

Keywords: Real-Time Service Oriented Architecture, Re-allocation resource, resource reservation management

1. INTODUCTION

By increasing web services size within dynamic environments, the requirement for more flexible resource reservation in distributed systems will be arises. In such systems, resource reservation means the request for required time periods of service form the resource for implementation. In Real-Time system, resource reservation is in way to guarantee resources by reservation indicators and implementation of each service within determined time period. By using appropriate database, access of service requests to resource free time will become quicker and easier. In order to allocate resource time before implementing service, databases are designed that can be divided into two linear and tree kinds. Three important factors of choosing appropriate database are: the aggregation time shall not be fixed, the allocated time interval to service shall be determined and the access to remained resource free time shall be possible [1].

Some of utilized tree structures in resource reservation management are mentioned in second resource [1]. hTBTree are usually implementing more appropriate than other kinds of structures [3], since, by entering service request, before reservation it gets assured whether the free time in requested time period is adequate or not? For this issue, the free time of parent nodes will be checked whether the required time of the service is less than registered free time in parents nodes or not? Since, free time of all parents' nodes shall be equal or more than requested time period of the registered service to implement service request. While, in hTBTree, it is only fulfilled through child node free time in order to allocate resource time. In other hand, dependency of whole allocation time according to the order of their entrance and having no solution for determining accurate allocated seconds from resource free time to service is considered as the deficiencies of binary trees [4]. Some utilized linear structures in resource reservation management have been introduced in resources [5,6].

**Corresponding author. Email address: emadi@iauyazd.ac.ir*

Special Issue: The Second National Conference on Applied Research in Science and Technology

Using tree structures due to having the ability of random access and different aggregation time in every level is more appropriate than linear structure. In tree structure, first total resource free time is replaced in root node and after that it will be replaced in children nodes.

Each node includes two fields [1]: Time period: [bottom limit (starting time) and top limit (End time)] and Current accessibility to resource. For instance, root node ([0,8],8) explains that the resource in this node will be reservation available from zero time to eight and the access to this resource in this node is 8 seconds, since no any allocation has been applied. In binary tree, each node explains half of time period of its parent. Each parent node free time is determined from sum of its left and right children free time. Different levels of tree indicate different aggregation time. Search for available time is started from root of tree and its survey is done from top to bottom. Ultimately, the leave that is completely matched with the duty will be chosen.

The purpose of this research is an attempt for more utilization from resource free time to accept more services and to increase the sum of resource free time reservation.

The structure of the paper is presented as follows: Section 2 introduces the tree structure, section 3, provides the proposed method, in section 4, the proposed method is implemented and evaluated, and finally in section 5, the conclusion of the discussions is presented.

2. TREE STRUCTURES

In this section, tree structures are described; because the ability of random access and different aggregation time in every level is more appropriate than linear structure.

2.1. Temporary Bınary Tree

Based upon Binary structure in this tree, all the free time of resource will be saved in root node and children nodes will explain the accessibility to resource free time in next levels. Each children node explains half of its parent time period. Each parent node free time is determined from the sum of its left and right children nodes free time. Different levels of tree indicate that aggregation time is different. Search for resource access soonest time is started from root and the survey is from top to bottom, when an appropriate node is found, its address will be stored for service insertion and its access is checked. If the resource accessibility in node is less than required amount, it will be displaced until this node is being left and right children of other node for finding adequate access to parent node. If current node is a root, it doesn't have adequate resource access and the service will be rejected [2].

2.2. hTB Tree

This tree is developed version of temporary binary tree. First total resource free time will be placed in root node and after that in children nodes, then by entering each service request in hTBTree, the requested service period will be mapped to minimum coverage set. Then search in requested service of minimum coverage set will be started. The first sub-time period that can fulfill whole or a part of service required time, will be allocated to the service. In case that the service haven't received all it required time, remain time will be allocated from next sub-time period. In case that, the available sub-time periods in minimum coverage set cannot allocate whole the required time for service, the service will be rejected [3].

2.3. JOB TREE

When a new service is rejected, it means that its requested time-period has been already reserved by other services. In order to avoid such problem, Job tree is utilized. When a new request is received, first the requested time period will be compared in the HtBTree with parent nodes. In case that requested time period of the new service is a sub-set of a node that its free time is zero, sub-time periods conflicts shall be determined. According to figure 1, sub time periods are divided into three kinds of A, P and N [4]:

N: The accepted service's sub-time period doesn't have conflict with the time period of new service.

P: The accepted service's sub-time period has conflict with the time period of new service to some extent.
A: The accepted service's sub-time period is located in boundaries of new requested service.

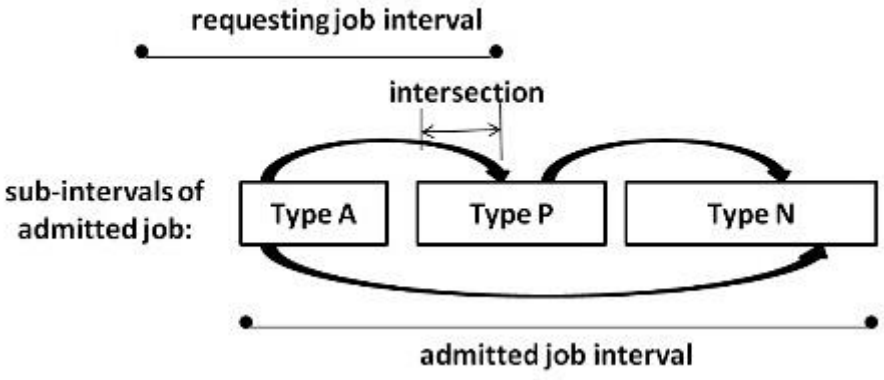

Figure 1. Three types of sub-intervals and reorganization [4]

Therefore, when a conflict happens, the time allocation in sub-time period of accepted service shall be turned into out of A or N kinds. In other words, first, the first sub-time period of minimum coverage set of accepted service that has conflict will be checked. Then, the required free time that shall be displaced is determined and the required time for accepted service will be fulfilled from other sub time periods. Therefore, sub-time period conflict with new service time period is resolved and the released time will be allocated to the new service. Since, determining kind of sub-time period is a time consuming task, in this research, it is tried to provide a method to access to remaining of resource free time and such time is utilized for re-allocation of services that have not received their whole required time.

3. PURPOSED METHOD

The purpose of Real- Time services is to implement service within requested time period and it doesn't mean to implement at the same of their arrival, therefore, it is tried to provide a solution to reserve resources for business process and input services to eliminate the risk of losing the time for each service. Resource reservation in provided solution includes below steps:

1. Time allocation and initial reservation of resource
2. Utilizing resource free time to re-allocate

In hTBTrees the allocated seconds from resource free time period are not clarified in non-leave nodes. This issue made resource free time re-allocation difficult and caused resource free

time reservation decrease [4]. In the recommended method, after entering services to system, they will be stored in the list based upon their top limit of their requested time period, then, initial resource reservation will be applied till it is reaching to last services including minimum top limit. By doing this action, resource free time will be allocated based upon the requirement of each service and the allocated seconds to each service will be clarified. During initial reservation, a time period from resource free time will be established that can be used in reservation of services that their total time period are not fulfilled. After each allocation, in case of meeting some conditions, established time periods will be utilized for re-allocation of resource free time in service. This issue causes reservation increase of resource free time and it will provide more flexible reservation.

3.1. Data Model

In recommended method, by using resource free time for re-allocation, resource free time reservation decrease which is one of reservation management deficiencies in method [4] has been improved. By each resource free time allocation, two kinds of periods will be established:
Allocation period (Alloc [i]): it indicates allocated time period to service request.
Resource free time period (Gap[i]): it indicates resource free time after allocation.
After each allocation, according to figure 2, a time period of resource free time will be created and stored in free time period vector, GAP[I].

GAP[i]:	GAP[1]	GAP[2]	GAP[3]	GAP[4]	GAP[5]	GAP[6]	GAP[7]	GAP[8]

Figure 2. Resource free time vector.

The steps of provided resource free time re-allocation in this research are as follows:

1. Function implementation is started from service or services that are reserved up to now, but they haven't gained all their required time.
2. Then it is checked whether resource free time re-allocation is possible or not?
3. If the resource free time period can cover the requested time period of service. (For instance, by allocating required time period of the service, new allocating time period does not excess top limit of gap and etc.) this time period will be utilized for resource free time re-allocation.
4. Possible amount of time period becomes another allocation for service i.
5. After allocation, the amount of resource free time shall be modified.
6. If allocation is not applied completely within resource free time period, required time remaining of service (x_i) and sum of requires time periods remaining of services shall be modified and update from the first allocation to present time.
7. If service doesn't have re-allocation during resource free time, the service will be rejected. Therefore, time units that are already reserved for initial allocation for service will return to resource free time status. Also, the sum of required times remaining of services will be modified from the first allocation to present time.
8. Ultimately, if resources free time period is utilized in a service re-allocation, time periods shall be modified. For this issue, service required remaining from the total resource free time and sum of required services of remaining time periods from the first allocation until now will be reduced.

4. RESULTS and DISCUSSION

Implementation is designed in MATLAB programming software. Input parameters of resource free time remaining and requested time period of a service which its whole required time period has not been fulfilled in initial allocation. Also, required time period of service and total amount of resource free time is also considered as other input parameters. Output includes allocated time periods from resource free time to each service that contains allocated time period during initial allocation and allocated period to service after re-allocation of resource free time.

The resuts of analysis is indicated in figure 3 and table 1. The amounts of reserve for each of six entered services to system are indicated in figure 3 by having 32 seconds resource free time. Blue bar column indicates ideal mode that the service has gained its required time period. Green bar column indicates reserved time of the service through hTBTree. Red bar column explains reserved time of service in recommended method.

As it is indicated in figure 3, in method of hTbTree, the service has not gained 17 seconds in period of [5, 22] its required time period and it is not accepted. In other words, from 32 seconds, 15 seconds is reserved and 17 seconds of resource time has remained unused. In recommended method, the requesting service is rejected in time period of [17, 32]. Therefore, from 32 seconds, 27 seconds of resource time is reserved and only 5 seconds of resource time is remained unused.

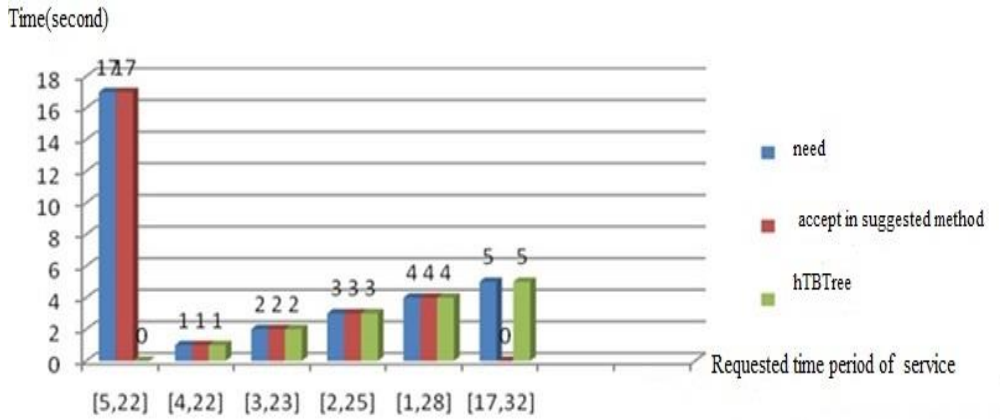

Figure 3. The comparison of service reservation acceptance through recommended method and provided method in [4].

Table 1. The comparison of resource time reservation in recommended method and hTBTree method.

Suggestion method	**hTBTree**	**method** / **Service**	
6	6	*Number of services*	***Resource free time =32 second***
[17,32]	[5,22]	*Requested time period of rejected service*	
5second	17 second	*Total unreserved time*	
27 second	15 second	*Total reserved time*	

5. CONCLUSION

In this article, a method has been presented that the reservation of resource free time is increased by using recommended method and using resource free time for re-allocation and more services will be accepted. Also, the feature of undetermined being of allocated time period

to each service is one of resource reservation deficiencies in method [3] which is improved. The results of implementation indicated that this method can allocate required time for more services and as a result, fewer services will be rejected.

REFERENCES

[1] M.Panahi, W.Nie, K-J.Lin.(2009),"A Framework for Real-Time Service-Oriented Architecture",on CEC '09 Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing,pp.460-467

[2] S.A.Moses, L.graunwald, and K.Dadachanji. (2008),"A scalable data structure for real-time estimation of resource availability in build-to-order environments ",Journal of Intelligent Manufacturing, pp.611–622.

[3] M.Panahi, W.Nie, K-J.Lin.(2009), "The Design and Implementation of Service reservations in Real-Time SOA", on ICEBE '09 Proceedings of the 2009 IEEE International Conference on e-Business Engineering, pp.129-136.

[4] S.zhou, k-j.lin.(2011), "A Flexible Srevice Reservation Scheme for Real-Time SOA", on 2011 eighth IEEE International Conference on e-business engineering, pp.215-222.

[5] Qing Xiong, Chanle Wu, Jianbing Xing, Libing Wu, Huyin Zhang.(2005), "A linked-list data structure for advance reservation admission control", Proceedings of the Third international conference on Networking and Mobile Computing, pp.901-910

[6] Tao Wang, Jianer Chen. (2002), "Bandwidth tree - a data structure for routing in networks with advanced reservations", in 21th IEEE International Conference on Performance, Computing, and Communications, pp.37-44.