2011 International Conference on Computational Intelligence and Communication Systems

A GA-based Approach to Resource Scheduling Supporting Flexible Quality Management of Ubiquitous Services

Mong-Fong Horng
Dept. of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan
Email: mfhorng@ieee.org

Yen-Ching Chan and Yau-Hwang Kuo
Dept. of Computer Sciences and Information Engineering, National Cheng Kung University, Tainan, Taiwan
Email: kuoyh@ismp.csie.ncku.edu.tw

Chia-Ming Yang and Jang-Pong Hsu
Advanced Multimedia Internet Technology, Inc.
Tainan, Taiwan
Email: jp@amit.com.tw

***Abstract*—In ubiquitous services, concurrent requests from various services for limited service resources such as network bandwidth, easily lead to a problem of resource insufficiency. The resource scheduling for ubiquitous services is the key to improve the tradeoff between request admittance, resource utilization and service quality. In this paper, a GA-based approach to resource scheduling to enable a flexible quality management of ubiquitous services is proposed to solve the problem mentioned above. First, the relationships between service of quality and resource requirements are explored. There are four different types of relations including (1) linear with saturation (LWS), (2) linear with deadzone and saturation (LWDS), (3) shifted step (SS), and (4) exponential (EX). Based on the derivation of the resource-quality model with the four relations, we define the maximum and minimum of resource requirement and regard the scope as the negotiation criterion for quality guarantee in genetic algorithm. Experimental results show that the proposed approach definitely benefits quality guarantee of service and the increasing of service request admittance ratio.**

***Index Terms*—Resource scheduling, Ubiquitous service, Genetic Algorithm, Quality of Service**

I. INTRODUCTION

IN a providing ubiquitous service environment[1], multi-users can request many services at the same time, and then it is likely to cause the condition of resource shortage [2-3]. Except for the absolutely sufficient resources, it is difficult to avoid the resource shortage. Therefore, the negotiation [4] among the all requested services and resource scheduling are important topics for discussion in realizing ubiquitous service. Because of limited resource, quality of service and the number of requests are also required to be considered in this research area. Usually, it is difficult to maximize the number of requests with the best quality guarantee. In ubiquitous environment, how to adaptively schedule the resource, particularly such as network bandwidth, is a significant issue to investigate [11]. Resource scheduling has been recognized as a method to find a resource composition between multi-layers. Thus, to find out the best solution, the considerable computation is necessary. The proposed GA-based solution benefits the reduction of computational cost and has a capability of searching near-optimal solution quickly.

The organization of the paper is as follows. In Section II, related work is introduced. We present the proposed service model, scheduling criteria and scheduling strategies in Section III. Simulation design and experimental results are shown in Section IV and the paper is concluded following a discussion in Section V.

II. RELATED WORK

In this part, we introduce and discuss with a few resource scheduling algorithms as follows. Opportunistic Load Balancing (OLB) [5-7] makes every machine keep a busy condition. This resources scheduling algorithm distribute the unassigned tasks to the available machines for the moment in arbitrary order, but it doesn't consider the workload of every machine at present. The best advantage of OLB resource scheduling algorithm is quite simple. As a whole, without consideration of the expected task execution time, the makespan of OLB takes much more time. Minimum Execution Time (MET) [5][8] resource scheduling algorithm has an intention to let every task can be supported by the best machine. MET scheduling algorithm will dispatch the unexecuted task to the machine which has the least execution time arbitrarily without taking account of the workload of the machines presently. This resource scheduling algorithm may result in the unbalance between all computers in the entire system so MET is not suitable to be applied with the heterogeneous systems.

Minimum Completion Time (MCT) [5-8] resource scheduling algorithm distributes the unexecuted task to the machine which has the minimum completion time at random. This scheduling algorithm calculates the completion time by adding ready time and execution time of each current task and assigns the task with the smallest completion time to the earliest available machine. But there are some tasks which can't get the minimum completion time. Minimum-Minimum Completion Time algorithm [5][7][9-10] calculates the minimum completion time of each unscheduled task and assigns the task to the machine which also has the minimum completion time. Thus it is called as Min-Min scheduling algorithm. The advantage of this algorithm is to consider the minimum completion time of all tasks, but it also spends more time on computing than other algorithms. Though the foregoing scheduling algorithms have the effects on some applications, the quality of service are not under consideration. As a result of the unconcerned factor, the system may provide a whole service but the users are not served with quality-guarantees services.

Bio-inspired Computing has been a new expertize to offer the effective and efficient search for optimal solutions of a defined cost function in the past decades. There were some paradigms such as genetic algorithm (GA), neural networks, particle swarm optimization, ant colony, immune system, and so on. These methodologies have been successfully applied on a diverse of target problems [12-15] to obtain fine solution.

978-0-7695-4587-5/11 $26.00 © 2011 IEEE
DOI 10.1109/CICN.2011.6

III. A Resource Scheduling Approach of Flexible Quality Management

A. Service Model

We assume a request can include just a service or a composition of services, a service only demands a device, and a device can support one service at least. The relations among requests, services and devices are shown in the below Fig. 1.

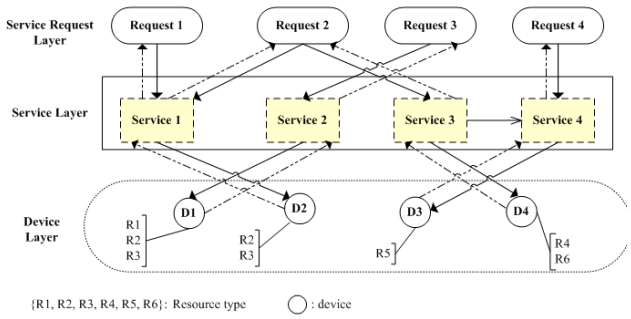

Fig. 1. The Relations among Requests, Services and Devices.

The service profile of ubiquitous service is predefined to describe the identity and the demanded resource of this ubiquitous service. S_i (where $i = 1, \dots, i_n$) is denoted a ubiquitous service provided by this ubiquitous service environment. i is denoted the identity of the ubiquitous service ,and there are i_n types of the ubiquitous services in all. Each element r_k^i (where $k = 1, \dots, k_n$) in S_i is one of the demanded resources of S_i. k is denoted the type of the resource ,and there are k_n types demanded resources provided by this ubiquitous environment. Each element r_k^i in S_i is composed of two units, $r_{k,min}^i$ and $r_{k,max}^i$. $r_{k,min}^i$ is denoted the minimum of the resource type k when the ubiquitous service i provides the lowest quality of service which the users can endure. $r_{k,max}^i$ is denoted the maximum of the resource type k when the ubitquitous service i provides the highest quality of service. Therefore, the value of r_k^i is between $r_{k,min}^i$ and $r_{k,max}^i$.

The device profile is predefined to describe the identity and the available resource of the device. D_j (where $j = 1, \dots, j_n$) is denoted a device provided by this ubiquitous service environment. j is denoted the identity of the device ,and there are j_n types of the devices in all. Each element R_k^j (where $k = 1, \dots, k_n$) in D_j is one type available resource that D_j provides.

The request profile is predefined to describe the identity and the required ubiquitous services. $Req_{x,y}$ (where $x = 1, \dots, x_n$, $y = 1, \dots , 4$) is denoted that a ubiquitous service request is required by a user. x is denoted the serial numbers of the required ubiquitous service requests in a time interval, and y is denoted the priority level of the ubiquitous service. Each $Req_{x,y}$ is composed of a ubiquitous service or ubiquitous service composition. Each element $S_{i,v}^{x,y}$ (where $v = 1, \dots, v_n$) in $Req_{x,y}$ is one of the required ubiquitous services. In a $Req_{x,y}$, v is denoted the serial numbers of the required ubiquitous services, and there are total v_n ubiquitous services which are required.

B. Scheduling Criteria

Different types of services have different relations between quality of service and resource. The relationship was rarely investigated in previous work. In this paper, we would like to propose several models to quantify this kind of relationship; including Linear with Saturation(LWS), Linear with deadzone and saturation (LWDS), Shift-Step (SS) and Exponential(EX).

First of all, supplying more resource will usually obtain better quality of service. But, the quality of service will reach the saturation when the offered resource exceeds the certain resource requirement. The relation between resource requirement and quality of service shown in Fig. 2 and can be represented by the linear equations. In this work, such type relation is called as Linear with Saturation (LWS).

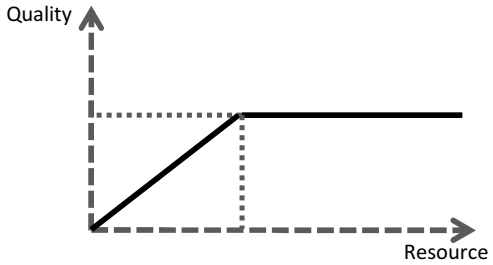

Fig. 2. The Relation of Quality and Resource is Linear with Saturation (LWS).

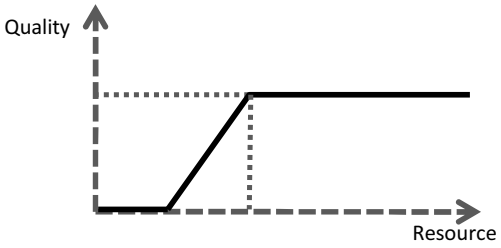

Fig. 3. The Relation of Quality and Resource is Linear with deadzone and saturation (LWDS).

Secondly, when the amount of resource does not exceed the certain resource requirement, the required request will not be able to be executed. Once reaching the lowest resource requirement, the later relation between resource requirement and quality of service is the same as LWS, and is shown in Fig. 3. So, in this work, such type relation is called as linear with deadzone and saturation (LWDS). Next, not only the amount of resource reaches the threshold value and then the service starts to serve, but also it will keep the same quality even if the system allocates much more resource. The relation resource requirement and quality of service is shown in Fig. 4, and is called as shifted step (SS) in this work. If more allocated resource still provides better quality of service, but it is different from LWS and LWDS. In this type of situation, the relation between resource requirement and quality of service is not linear but exponential. This type of relation is shown in Fig. 5 and is called as exponential (EX) in this work.

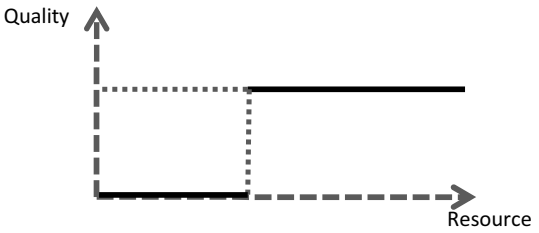

Fig. 4. The Relation of Quality and Resource is Shifted Step (SS).

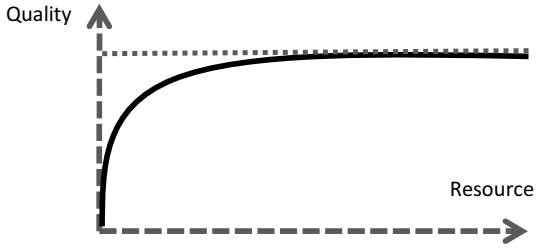

Fig. 5. The Relation of Quality and Resource is Exponential (EX).

Therefore, through the foregoing relations among service of quality and resource requirement, we can define a range for each relationship and the Quality Funtion as follows:

Definition 1: Quality Function

We define that $q = Q(r)$, and the formula depends on which kind relation the ubiquitous service has.

- Linear with Saturation (LWS)

$$Q(r) = \begin{cases} 0, & r = 0 \\ \frac{R_U - r}{R_U}, & 0 < r < R_U \\ 1, & r \geq R_U \end{cases} \quad (1)$$

- Linear with Deadzone and Saturation (LWDS)

$$Q(r) = \begin{cases} 0, & r < R_L \\ \frac{r - R_L}{R_U - R_L}, & 0 \leq r < R_U \\ 1, & r \geq R_U \end{cases} \quad (2)$$

- Shifted Step (SS)

$$Q(r) = \begin{cases} 0, & r < R_L \\ 1, & r \geq R_L \end{cases} \quad (3)$$

- Exponential (EX)

$$Q(r) = 1 - e^{-ar} \quad (4)$$

The proposed idea of resource scheduling is to obtain tunable ranges of resource requirements based on the resulted quality of service under the resource constraints. And we propose a mechanism ensure admitted requests with quality guarantees.

C. Scheduling Strategy

Our purpose is to admit more requests with quality guarantees. According to the different concerned factors, we design three scheduling strategies as the followings. Based on Section III, the equations are defined.

1) Minimum Quality Guarantee Resource Scheduling (MQGRS)

In this scheduling strategy, to admit the maximum number of requests is mainly taken into account. The objective function is formulated as bellow:

Maximize

$$\sum_{x=1}^{x_n} \beta_x \quad (5)$$

where

$$\beta_x = \begin{cases} 1, & Req_x \text{ is admitted;} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

2) Flexible Quality Guarantee Resource Scheduling (FQGRS)

In this strategy, we obtain a quality scope from the above mentioned four relations. Based on Definition 1, $\boldsymbol{Q(r)}$ of each service request can be calculated and each service request will have a quality value. We will find the service request set with the maximum sum of quality values for admitting. The objective function is formulated as bellow:

Maximize

$$\sum_{x=1}^{x_n} \sum_{v=1}^{v_n} Q(r_k^{xiv}), \forall i = 1, \dots, i_n; k = 1, \dots, k_n$$

subject to

$$\sum_{x=1}^{x_n} \sum_{v=1}^{v_n} r_k^{xiv} \gamma_{xivj} \leq R_k^j \quad (7)$$

where

$$\gamma_{xivj} = \begin{cases} 1, & D_j \text{ is selected to } S_i \text{ in } Req_x; \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

3) First In First Out Resource Scheduling (FIFORS)

This scheduling strategy is the most traditional way without any rules but time. As the literal meaning, the earliest required service request will be executed at the first time in the duration and so on until the waiting queue is empty.

4) Priority Resource Scheduling (PRS)

In this scheduling strategy, we defined four levels of priority for all services. They are level-1, level-2, level-3 and level-4. Every ubiquitous service request is set up with a parameter. The parameter denotes which priority level the service has. For example, the parameter of the ubiquitous service request is level-1, and this request will be given precedence for admitting. When the requests with level-1 are finished scheduling, the system will start to schedule the requests with level-2 and so on.

D. The Design of GA Algorithm

In order to keep the diversity among all chromosomes, the initialization population is produced randomly. We use 12-bit stings to represent the value between 0 and 1. The value is $Q(r)$, which is the allocated quality of a service request. The fitness function is defined to evaluate the possibility of a chromosome whether it is the best solution. In this paper, we design two fitness functions to find the feasible solutions for MQGRS and

FQGRS. The fitness functions of MQGRS and FQGRS are shown respectively as bellow

$$fitness(\beta) = \sum_{x=1}^{x_n} \beta_x \quad (9)$$

$$fitness\big(Q(r)\big) = \sum_{x=1}^{x_n} \sum_{v=1}^{v_n} Q\big(r_k^{xiv}\big) \quad (10)$$

We adopt the Roulette Wheel Selection as our selection mechanism. And a chromosome with higher fitness value will have the higher possibility to be selected. Genetic operators, which are crossover and mutation, are used to generate the next generation. These two operators are described as follows,

1) **Crossover**
   A crossover operator that selects a crossover point randomly within a chromosome then combines the two parent chromosomes at this point to produce two new offspring. A multi-point crossover is adopted here.
2) **Mutation**
   A typical mutation is to select a single bit in a chromosome and flip it. This means that if the selected bit is a 1 it now becomes a 0 and vice versa. Mutations have to occur rarely, or they will result in that it is hard to converge toward an ideal value for fitness.

In this paper, we specify the termination condition of 500 generations to evolve those chromosomes for the optimal solution.

IV. Numeric Results and Analysis

In this chapter, some experiments are designed and made to observe the performance of Flexible Quality Guarantee Resource Scheduling (FQGRS). The experimental results are also compared with other scheduling strategies such as Minimum Quality Guarantee Resource Scheduling (MQGRS), Priority Resource Scheduling (PRS) and First In First Out Resource Scheduling (FIFORS). The construction of a ubiquitous service environment needs to consider many phases and integrate a lot of different devices. On the other hand, the establishment of an ideal ubiquitous environment is a very complicated work. The experimental goal is to express that the flexibility of the proposed resource scheduling. Therefore, we narrow down the experimental scope and simplify the experimental environment.

We emulate a single resource-constrained situation so take bandwidth-constrained situation as the case study. First, we assume a small-sized enterprise network offering broadband network service as our experimental environment. There are four typical kinds of network services used in enterprises. These services are streaming service that may often be used to hold the video conferences, VoIP service that are used not only in video conferences but also instead of classic phones probably, FTP service that are used to transmit any type of files, and browsing web page service. In this experiment, every request is only composed by one service. We will select one of those provided network services, and increase the number of selected network service. Through the increasing number of network services, it can make different overloading situations.

If the value of the overloading ratio is a positive number, it will be shown in the field. Otherwise, it will be shown as "Non-overloading" instead of the negative number.

We choose to increase the number of browsing web page service in order to emulate various situations from non-overloading to overloading. Then, dealing with each situation uses those mentioned resource scheduling methods, FQGRS, MQGRS, PRS, and FIFORS. Fig. 6-8 show the bandwidth allocations for all service requests in case of 20, 30, 40 service requests. This allocation is done by the proposed FQGRS. And x and y axes denote the serial numbers of total required requests in the duration and the amount of bandwidth respectively. The number of x axis represents one request and the maximum of x axis also means the total amount of required requests in the duration. The broken line with squares represents the maximum of required bandwidth of one request. The broken line with hallow circles represents the minimum of required bandwidth of one request. The broken line with pentagrams represents the actual amount of allocated bandwidth of each request.

As shown in Fig. 6-8, the line with pentagram is always between maximum and minimum bounds. The pentagram sign also means that the required request is admitted. When the total requests are admitted in the duration, the allocated bandwidth is between the minimum bandwidth and maximum bandwidth. In Fig. 6, FQGRS admits all the required requests. The height of the broken line with pentagrams in Fig. 6 is lower because FQGRS sacrifices the quality of each request to save more bandwidth for allocating resource to other requests. In Fig. 8, the line with pentagrams has two breaks that mean FQGRS drops two requests. The overloading situation makes FQGRS not only sacrifice the quality but also drop the certain requests which need more required resource for admitting other requests which need less required resource. In Fig. 9, with higher overloading, there are more and more breaks on the broken line with pentagrams and the height of the line with pentagrams is getting lower and lower but still above the minimum of required bandwidth. As a result, FQGRS not only enhances the number of admitted

V. Conclusion and Future Work

In this paper, we propose a flexible resource scheduling method for the resource-limited problem in ubiquitous environment, called Flexible Quality Guarantee Resource Scheduling (FQGRS). The experimental results show that FQGRS looks after both sides that are the number of admitted requests and the quality of each admitted request. In an overloading situation, FQGRS is able to admit much more requests with guaranteed quality which is between the minimum and maximum of quality. In the future, we can give careful consideration to the distribution of required service request sequence to establish a coping mechanism. With different distribution, this mechanism can set different order of adjustment in the duration to make FQGRS has more delicate flexibility. Moreover, we also take account of the size of the tunable range and the relation of quality and resource at the same time to find out the more applicable order of adjustment to improve FQGRS.

ACKNOWLEDGEMENT

The authors would like to thank sincerely the partial financial support from National Science Council, Taiwan under the research projects with grants of 97-2221-E-006-144-MY3 and 98-2221-E-151-029-MY2.

REFERENCES

[1] M. Weiser, "The Computer for the 21st Century," Human-computer interaction, Morgan Kaufmann Publishers Inc. pp. 933-940, Sept. 1995.

[2] H. A. Schmid, "Service Congestion: The Problem, and an Optimized Service Composition Architecture as a Solution," in Proceedings of IEEE International Conference on Web Services, pp. 505-514, USA. 2006.

[3] M. P. Singh, "Trustworthy Service Composition: Challenges and Research Questions," in Proceedings of the Autonomous Agents and Multi-Agent Systems Workshop on Deception, Fraud and Trust in Agent Societies, pp. 30-38, July 2002.

[4] T. Sun, S. Li, and Q. Zhu, "An Optimized Strategy of Service Negotiation," in Proceedings of the Second IEEE International Symposium on Service-Oriented System Engineering, pp. 210-214, USA, 2006.

[5] R. Armstrong, D. Hensgen, and T. Kidd, "The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-time Predictions," in Proceedings of 7th IEEE Heterogeneous Computing Workshop, pp. 79-87, 1998.

[6] R. F. Freund and H. J. Siegel, "Heterogeneous Processing," IEEE Computer, Vol. 26, No. 6, pp. 13-17, Jun. 1993.

[7] R. F. Freund, et al., "Scheduling Resources in Multi-User, Heterogeneous, Computing Environments with SmartNet," in Proceedings of 7th IEEE Heterogeneous Computing Workshop, pp. 184-199, Mar. 1998.

[8] T. D. Brauna, et al., "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems," Journal of Parallel and Distributed Computing, Vol. 61, Issue 6 , pp. 810-837, 2001.

[9] O. Ibarra and C. E. Kim, "Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors," Journal of the ACM, Vol. 77, No. 2, pp. 280–289, Apr. 1977.

[10] G. Ritchie and J. Levine, "A Fast, Effective Local Search for Scheduling Independent Jobs in Heterogeneous Computing Environments," Journal of Computer Applications, Vol. 25, Issue 5, pp. 1190-1192, 2005.

[11] M. F. Horng, Y. H. Kuo, L. C. Huang, Y. T. Chien," An effective approach to adaptive bandwidth allocation with QoS enhanced on IP networks," In Proceedings of ACM International Conference on Ubiquitous Information Management and Communication, pp. 260-264, Korea, 2009

[12] P. Puranik, P. Bajaj, A. Abraham, P. Palsodkar, and A. Deshmukh, "Human Perception-based Color Image Segmentation Using Comprehensive Learning Particle Swarm Optimization", Journal of Information Hiding and Multimedia Signal Processing, Vol. 2, No. 3, pp. 227-235, July 2011.

[13] K. Loukhaoukha, J. Y. Chouinard, and M. H. Taieb, "Optimal Image Watermarking Algorithm Based on LWT-SVD via Multi-objective Ant Colony Optimization," Journal of Information Hiding and Multimedia Signal Processing, Vol. 2, No. 4, pp. 303-319, October 2011.

[14] M. F. Horng, Y. T. Chen, S. C. Chu, J. S. Pan, B. Y. Liao, "An Extensible Particles Swarm Optimization for Energy-Effective Cluster Management of Underwater Sensor Networks," in Proceedings of International Conference on Collective Computational Intelligence, LNCS6422, Springer, pp. 109-116, Taiwan, 2010.

[15] H. Al-Qaheri, A. Mustafi, and S. Banerjee, "Digital Watermarking using Ant Colony Optimization in Fractional Fourier Domain, "Journal of Information Hiding and Multimedia Signal Processing, Vol. 1, No. 3, pp. 179-189, July 2010.

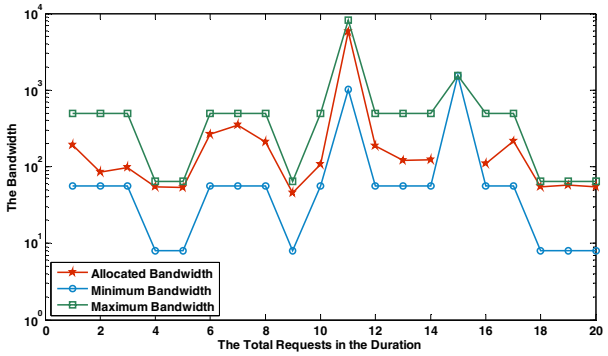

Fig. 6. The Allocated Resource of Each Admitted Request in case of 20 requests 2

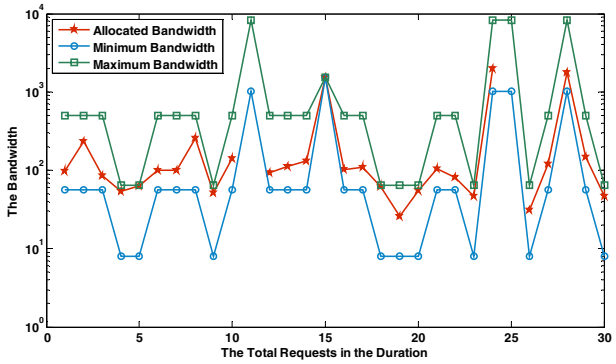

Fig. 7. The Allocated Resource of Each Admitted Request in case of 30 requests.

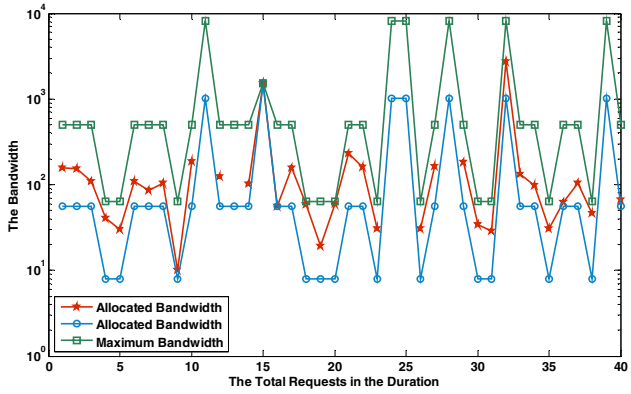

Fig. 8. The Allocated Resource of Each Admitted Request in case of 40 requests.