IJICTR International Journal of Information & Communication Technology Research

Volume 7- Number 3- Summer 2015 (53-61)

Resource Reservation in Grid Networks based on Irregular Cellular Learning Automata

Sara Rezaei
Department of Computer Engineering
Islamic Azad University-South of Tehran Branch
Tehran, Iran
Sara.rhezaii@gmail.com

Ahmad Khademzadeh
Iran Telecom Research Center
Tehran, Iran
zadeh@itrc.ac.ir

Mansour Sheikhan
Department of Electrical Engineering
Islamic Azad University-South of Tehran Branch
Tehran, Iran
msheikhn@azad.ac.ir

Received: March 13, 2015-Accepted: June 4, 2015

Abstract— **Computing infrastructures that are based on grid networks have been recognized as a basis for new infrastructures of distributed computing. Providing appropriate mechanisms for scheduling and allocating resources to user's requests in these networks is considered very important. One of the current issues in the grid networks is how to ensure the precise timing of executing requests sent by users, especially requests that have deadlines and also co-allocation requests. The resource reservation has been mainly developed to address this problem in the grid systems. On the other hand, models based on the cellular automata have advantages such as lower processing complexity, configurability of the cells, and the ability of predicting future conditions. In this study, an efficient model based on irregular cellular learning automata (ICLA) is presented for the task of resource reservation. The proposed model was simulated on a network with random topology structure. The performance of proposed method was compared with two well-known algorithms in this field. The experimental results showed increased efficiency in the resource utilization, decreased process execution delays, and reduced rate of request rejection.**

Keywords- grid computing; advance reservation of resources; resource allocation; irregular cellular learning automata; Job scheduling.

I. INTRODUCTION

In the past decade, the grid technology has been developed tremendously and has made its appearance as a significant achievement in addressing large scale compute-intensive or data-intensive applications [1]. In other words, the grid technology has been proposed to optimize the utilization of distributed resources and to provide the possibility of sharing these resources in a controlled manner. In essence, grid is "a type of parallel and distributed system which provides the ability of sharing heterogeneous resources and also provides a dynamic and runtime-based allocation of resources to request/ application with respect to resource availability, cost, efficiency, capacity and quality of service requirements" [2]. Thus, it is clear that one of the basic needs in these networks is the optimal utilization of resources to increase the efficiency and to reduce the execution time of an application. A system known as scheduler is very essential and important in making better use of resources in a grid environment. Schedulers manage the distribution and allocation of

processor resources available in the system, among the requests sent by users [2, 3]. One of the important aspects evaluated in using network resources is the quality of service. Quality of service guarantees that the system is providing quality requirements for using the resources. One way to ensure quality of service is the advance reservation of required resources, which makes sure that the resources are available when needed and requested [4-6]. Various researches have already used this technique to increase the speed and to reduce the response time. Advance reservation of resources has been used as an effective method for supporting quality of service in many grid systems. Advance resource reservation allows applications to have simultaneous access to the resources and also ensures the availability of resources when needed. But advance reservation can also have negative effects on the sharing resources and the scheduling tasks in grid systems. For example, studies have shown that fixed reservation causes resource fragmentation and low rates of resource utilization, and that the excess reservation often causes high reject rate [7].

In this study, given the configurability of cellular automata-based models and their ability in predicting future conditions, we provided a new mechanism for resource reservation (advance resource reservation) in the grid networks based on irregular cellular learning automata (ICLA). The model proposed in this paper is divided into three phases: (a) advance reservation of resources, (b) allocation of processes which are going to be sent to the resources, (c) scheduling the processes of each resource. Phases (a) and (b) of the model are based on irregular cellular learning automata structure and have been designed with the aim of improving the algorithm behavior and increasing the efficiency of resources by receiving feedback from the environment. The third phase is based on an innovative approach and tries to prioritize resources to reduce the processing delays and the number of users' requests rejection.

This paper is organized as follows: we will review the previous works related to this area in the second section. In the third section, we will explain the model of irregular cellular learning automata. In the fourth section, we will present the proposed model and after that, we will evaluate the proposed method and the simulation results will be presented. In the last section, we will present the conclusions and future works.

II. RELATED WORK

In the last decades, various studies have introduced and classified algorithms and methods for managing resource allocation requests. Accordingly, the techniques used for managing resource allocation requests, which include scheduling and advance resource reservation, are also taken into consideration in different aspects [8, 9].

Smith et al. have evaluated and examined the issue of providing the ability of advance reservation in a supercomputer-based computational resource by the use of different models [10].

The backfilling method was implemented for the first time in the Argonne National Laboratory for IBM SP2 supercomputer systems on Extensible Argonne Scheduling system (EASY) local scheduling system. The backfilling technique is considered as one of the most popular methods for improving the utilization rate of reservation-based resources. Generally, the backfilling technique is classified into two groups: Aggressive Backfilling Reservation Policy (ABRP) and Conservative Backfilling Reservation Policy (CBRP) [11]. However, the cellular automata-based methods have been considered for scheduling requests in the grid network in the recent decade [12, 13]. For example, Yeh and Wei [14] have presented an economic resource allocation model to obtain the service reliability of grid computing networks through Cellular Automata Monte Carlo Simulation (CA-MCS). A new cellular automata-based algorithm was proposed for scheduling independent tasks with aim of optimizing the time in economic computational grids [13]. An evolutionary cellular automata-based scheduling method based on genetic algorithm was proposed in another study for scheduling parallel processing in multi-processor systems [15]. Another study presented and evaluated the learning automata-based algorithms called LATO and ALATO for scheduling the tasks with the aim of minimizing the execution time of applications [16]. An Overlapped Advance Reservation Policy (OARS) was introduced to reduce the negative effects of advance allocation of resources (advance reservation), in a way that allowed a new reservation request that overlapped with current reservations be accepted in the system under certain circumstances. This policy shows more compliance when grid systems have high reservations rates [17]. In addition, several algorithms have been proposed for resource reservation in the grid networks; for example refer to [18-20].

III. IRREGULAR CELLULAR LEARNING AUTOMATA (ICLA) MODEL

The ICLA overcome the restrictions of rectangular grid structure in traditional cellular learning automata. This is important, because there are many different applications (such as wireless sensor networks, open network systems, and graphs related applications) that cannot be modeled properly with the rectangular grids. The cellular learning automata are defined as nonlinear graphs, where each vertex represents a cell equipped with a learning automaton (LA) [21, 22]. Each LA, located in a particular cell, determines its state or action based on its action probability vector. Similar to the cellular learning automata (CLA) [23], there is a rule that ICLA operates based on it. The rule of CLA and the actions selected by the neighboring LA of any particular cell, create a specific reinforcement signal for the learning automata located in that cell. In fact, learning automata neighboring (adjacent to) that particular cell (learning automaton) form a local environment for that cell. The local environment of a cell, is non-stationary and variable in a way that the action probability vectors of neighboring learning

automata change during the evolution of irregular cellular learning automata [21] (Fig. 1).

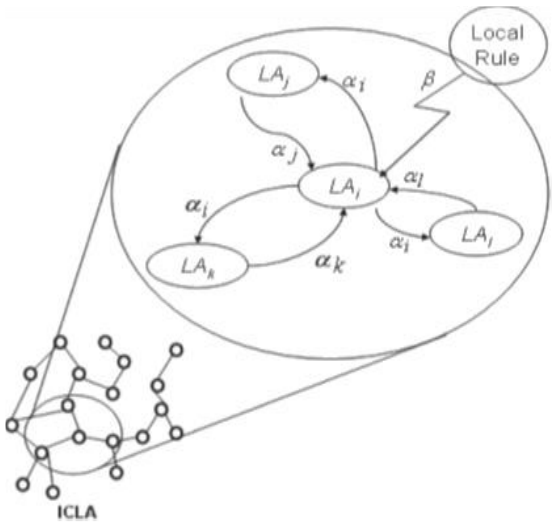


Fig1. Structure of irregular cellular learning automata [21]

In other words, the learning algorithm used in this model, which is a linear learning algorithm, is as follows:

Suppose that the action α_i is selected in n^{th} step:

- Optimal response:

$$p_i(n+1) = p_i(n) + \alpha\left[1 - p_i(n)\right] \quad (1)$$

$$p_j(n+1) = (1-\alpha)p_j(n) \quad \forall j \quad j \neq i \quad (2)$$

- Non-optimal response:

$$p_i(n+1) = (1-\beta)p_i(n) \quad (3)$$

$$p_j(n+1) = (\beta/r-1) + (1-\beta)p_j(n) \quad \forall j \quad j \neq i \quad (4)$$

where α and β are reward and penalty parameters. When $\alpha=\beta$, automaton is called L_{RP}. If $\beta=0$ and $0<\beta<<\alpha<1$, the automaton is called L_{RI} and $L_{R\varepsilon P}$, respectively.

IV. PROPOSED MODEL

In order to describe the proposed model for resource reservation, the assumptions of this model are presented in the following subsections:

A. Assumptions

In this study, we assumed that in a grid network with random topology structure, resource set is denoted by $R = \{r_1, r_2, r_3, \ldots, r_n\}$, so that each resource such as r_i has a computing power equal to c_i (MI/sec) .

In other words, each ri resource can execute $c_i \times 10^6$ instructions per second. On the other hand, in this problem there are resource reservation requests that each user sends to network in order to execute its desired processes in a certain time period in the future which is shown with $RR= \{rr_1, rr_2, rr_3, \ldots rr_n\}$ and each reservation request has a start time, end time, required set of resources, minimum required computing power, and maximum required computing power. Also in this problem, there are m numbers of different processes which are represented by $P = \{P_1, P_2, P_3, \ldots, P_n\}$ and each process has the following characteristics:

1. Each process like P consists of O_i MI different instructions. In other words, the task or process P consists of $O_i \times 10^6$ different instructions which are shown by P_i.

2. Also for each process P, there is an entry time, which is shown by P_e, and a processing start time shown by P_s, and a finishing time represented by P_f. Also execution time of process is denoted by P_r.

3. In this model, the processes are divided into two categories: immediate and non-immediate in terms of constraints at run time (having a deadline). For each process of $p \in P$, which is in immediate form, we have: $0< P_d <\infty$.

4. The processes sent to network can be executed in two forms: with reservations and without reservation.

B. Structure of Proposed Model

As mentioned in assumptions, there are three main challenges in this problem:

1. Resource reservation

Users can reserve resources based on their own estimation of processes that they are going to send to network during each cycle. Therefore, each user sends its desired reservation request before sending its processes to the network and reserves resources for a certain period of time in the future.

2. Allocation of processes to resources

At this stage, each process has to be attributed to a specific resource so that the specified resource would be tasked with executing the instructions of that process. Resource reservation must be in a way that can ensure the execution of process and its instructions before the deadline.

If a process cannot be allocated to a specific resource, that process is assumed to be rejected. The resource reservation should be in a way that can reduce the request rejection rates and increase the exploitation and utilization of resources in system.

3. Scheduling processes in each resource

After assigning different processes to each resource, that resource is tasked with scheduling and prioritizing those processes. This scheduling must be in such a way that minimizes the processing delay. Scheduling processes must be carried out with respect to their entry time and their number of instructions and also the deadline of each process, so that all the processes would finish before the deadline and with a minimum delay.

In order to implement the first and the second phase (i.e., resource reservation and allocation of processes), a comprehensive algorithm should be implemented to

consider all resources and processes and then allocate the reservation requests and different processes to appropriate resources with respect to overall situation of that time. In the third phase or the stage of processes scheduling, resources can schedule the processes in parallel and separate form. It seems that more efficient output of the first and the second phases will lead to further improvement in the output of the third phase.

In other words, resource reservation plays an essential role in solving this problem, and if we manage to refer the processes to more appropriate resources, we can reduce the time period of waiting in the queue and ultimately the time of process execution, through better scheduling. We will describe three proposed algorithms in the following.

C. Proposed Algorithms

C.1. Advance Resource Reservation Algorithm

In the reservation phase, the objective is to allocate a resource to a reservation request in a way that resource would be appropriate to the characteristics of that request [24].

Each reservation request includes: start time, end time, the number of resources, the minimum computing power, and the maximum computing power.

The purpose is to give those resources to reservation requests that are more compatible with the type of request and also would reduce the request rejection rate. The steps of the algorithm used for achieving the objectives of this phase are given in Table 1.

In this algorithm, in each moment of T, the probability of selecting each neighbor is calculated by:

$$P_T(r_1) = p(r_1) \quad (5)$$

where $p(r_1)$ is the final calculated probability of sending a reservation request from r_1 resource to r_1 neighbor, and also we used Eqs. (6) to (9) in different steps of Algorithm 1:

$$P(r_i) = P_T(r_i) + a\left(1 - P_T(r_i)\right) \quad (6)$$

$$P(r_j) = P_T(r_j) - a\,P(r_j) \quad (7)$$

$$P(r_i) = (1-b)P_T(r_i) \quad (8)$$

$$P(r_j) = [b/(r-1)] + (1-b)P(r_j) \quad (9)$$

In the considered model, the probabilities of transferring a process from one resource to another neighboring resource are considered and these probabilities will change based on obtained results. For each resource, "r" represents the number of neighbors of that resource.

Learning algorithm is stopped after performing several runs and improving answers. Here ε is the parameter of acceptable error. If after executing several phases, the amount of improvement in algorithm is less than ε, then it is assumed that the algorithm has reached the appropriate answer.

In the above equations, "a" is the reward parameter and "b" is the punishment or penalty parameter. If a = b, then reward and penalty values for learning algorithm are equal and this algorithm is called L_{R-P}. If b<<a, then the algorithm is called $L_{R\varepsilon P}$ and if b = 0, then the algorithm is called L_{R-I}.

C.2. Algorithm of Allocating Resources to Processes

Table1. Algorithm of reserving resources for each reservation request

1	**Input**: specifications of each reservation request
2	**Output**: specifications of resources allocated to each reservation request
3	For each $rr \in RR$ request, run input
4	Per rr_r run
5	If no resource is free in the period $[rr_s, rr_f]$ that $rr_{min} < r_i < rr_{max}$ then
6	Consider the reservation request as rejected.
7	Else
8	Randomly select one of the resources and assign the reservation request to that
9	Set $e_{old} = \infty$
10	Set $e_{new} = \lvert r_I - r_{min} \rvert$
11	Until $\lvert e_{old} - e_{new} \rvert \geq \varepsilon$
12	Select one of the resource neighbors, such as r_2 by Eq. (5) that: $rr_{min} < r_{2I} < r_{max}$
13	$e_{old} = e_{new}$
14	$e_{new} = \lvert r_{2min} - r_{2I} \rvert$
15	If $e_{new} < e_{old}$ then
16	Increase the probability of selecting r_2 neighbor by Eq. (6) and set the probability of selecting other neighbors by Eq. (7).
17	Else
18	Set the probability of selecting r_2 neighbor by Eq. (8) and set the probability of selecting other neighbors by Eq. (9).

A method similar to the one used for resource reservation is utilized to solve the problem of resource allocation. If the user has already reserved the resource for its process, then the set of available resources is the same as specified in the previous

Table 2. Algorithm of resource allocation for each new non-reserved entering process

1	**Input**: specification of each entering process
2	**Output**: specifications of resources allocated to each process
3	For each $p \in P$ request, run input
4	If among the resources available for processing, no resource is free in the time period $\left[t, t+\frac{p_i}{r_I}\right]$ that $\frac{p_i}{r_I} < p_d$ then
5	Consider the run request as rejected.
6	Else
7	Randomly select one of the resources and allocate the request to that
8	Set $e_{old} = \infty$
9	Set $e_{new} = \left\|\frac{p_i}{p_d} - r_I\right\|$
10	Until $\left\|e_{old} - e_{new}\right\| \geq \varepsilon$
11	Select one of the resource neighbors, such as r_2 by Eq. (5) that: $\frac{p_i}{r_{2I}} < p_d$
12	$e_{old} = e_{new}$
13	$e_{new} = \left\|\frac{p_i}{p_d} - r_{2\min_I}\right\|$
14	If $e_{new} < e_{old}$ then
15	Increase the probability of selecting r_2 neighbor by Eq. (6) and set the probability of selecting other neighbors by Eq. (7).
16	Else
17	Set the probability of selecting r_2 neighbor by Eq. (8) and set the probability of selecting other neighbors by Eq. (9).

C.3. Algorithm of Scheduling Processes in Each Resource

After identifying the resources responsible for addressing each process and sending the processes to their specified resource, the processes will be placed in a waiting queue and a scheduler should select processes one at a time from that queue and send them to resource for execution.

The steps of the algorithm used for selecting the process from the waiting queue of each resource are given in Table 3.

V. Simulation Results and Performance Evaluation

In this section, we will present the simulation results and evaluation of the proposed model. MATLAB software was used to simulate the proposed model.

Table 3. Algorithm of prioritization of processes in the waiting queue of each resource

1	**Input**: a list of processes available in the waiting queue of each resource
2	**Output**: priority of execution of processes available in the waiting queue of each resource
3	For each process $p \in P$ that is in the waiting queue of resource R, calculate the value t_d for the new process by the following equation: $t_d = (p_e + p_d) - T$
4	If $t_d < \frac{p_i}{r_I}$ then
5	Exclude process P from the waiting queue and consider it as undone
6	Else
7	Set the priority of process to $-1 \times \left(t_d = \frac{p_i}{r_I}\right)$
8	If p_1 process is running and a process such as p_2 is in the waiting queue that $(p_{2e} + p_{2d}) - \frac{p_{2i}}{r_I} < (p_{1e} + p_{1d}) - \frac{p_{1i}}{r_I}$, then
9	If $\frac{p_{2i}}{r_I} + \frac{p_{1i}}{r_I} < p_{1e} + p_{1d}$, then
10	Place p_1 process in the waiting queue and run p_2 process
11	Else, if p_1 process is not reserved and p_2 is reserved, then
12	Place p_1 process in the waiting queue and run p_2 process
13	Else
14	Place p_2 process in the waiting queue and sort them according to priority
15	If no process is running
16	Select the process with the highest priority parameter value from the waiting queue and run it.

A network topology was created randomly (uniform random distribution) with "n" resources and "S" users, then "x" resource reservation requests were sent randomly by users to the network. In the next phase, users randomly sent m processes to the network for execution. For each part, the simulations were repeated at least 100 times and the results displayed in each part were the average values of the results obtained in 100 times of executing the algorithm for different scenarios with different topologies and structures.

The value of reward and penalty parameters for irregular cellular learning automata was set equal. In this environment, 12 powerful resources were considered for running user processes and the number of system users was 25, and each of them applied 50 to 80 processes. The number of sub- instructions of each process was set randomly in the interval [4200,5600] based on the uniform random distribution. By default, the resources were set to be available and unfailing throughout the whole simulation. In some scenarios

which we assumed the failure of a resource, that resource became unavailable for a maximum duration of 850 seconds (Fig. 2).

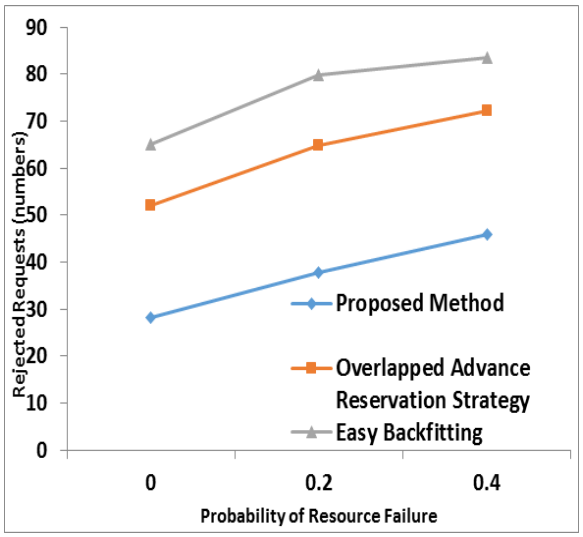


Fig 2. Effect of probability of resource failure on rejected requests

The simulation of proposed model was conducted based on mentioned variables and parameters, and then was compared with two well-known algorithms in this field called Easy Backfilling and Overlapped Advance Reservation Strategy (OARS) in terms of three parameters: resource utilization, the number of request rejection, and the process execution delays. Simulation results are presented in Figs. (3)-(5).

According to the obtained results, it can be seen that the proposed method has shown better performance than the other two methods. It should be noted that Easy Backfilling algorithm is based on FCFS and can be considered an improved version of that algorithm. In the Easy Backfilling algorithm, processes are prioritized based on shadow time as well as entry time. The OARS method tries to provide the capability of run time-based reservation of new requests and running new processes in the future, by estimating the run time for the processes and excess reservation required for each process. But, the solution proposed in this paper uses ICLA so that the system receives feedback from the environment and status of resources in the network and tries to improve its behavior and decisions regarding the allocation of reservation or job requests to the selected resources.

Furthermore, this feature also provides the ability of predicting future status of resources in the network which is an advantage over other two investigated methods. As is clear from the evaluation results, in comparison with other methods, the proposed solution has increased the efficiency of resources because of using the learning phase in algorithms given in Tables 1 and 2, so that at any given time the scheduler is aware of resources' status, and the time of their reservations and the amount of work that is in progress, and sends the job requests to resources that are free in users specified time frames by considering the current status and the reservations periods for each resource. Moreover, this solution uses the algorithm given in Table 3 and also considers the two factors of deadlines and resource reservation to prioritize all processes in the waiting queue of each resource in order to reduce the number of rejected requests and delays in the running processes; while the other two methods just use requests entry time and run time to prioritize the execution process. Therefore, the proposed solution has achieved better results than the two other competitive investigated methods (Figs. 3-13).

VI .Conclusions and Future Work

In this paper, we tried to provide a new and efficient solution for the issue of resource reservation in grid networks. The proposed model is based on irregular cellular learning automata (ICLA). We compared the proposed method with two well-known algorithms of Easy Backfilling and Overlapped Advance Reservation Strategy (OARS) in order to evaluate the provided approach.

The evaluation results indicated an increase in the efficiency of resource utilization, a decrease in the process execution delays and also a decrease in the request rejection rate for the proposed method compared with other two methods. In addition, given the capabilities of cellular learning automata (CLA), we hope to implement this method for distributed systems, and also to reach a resource allocation configuration that enables processes to be run partly on one resource and partly on other available resources in that network.

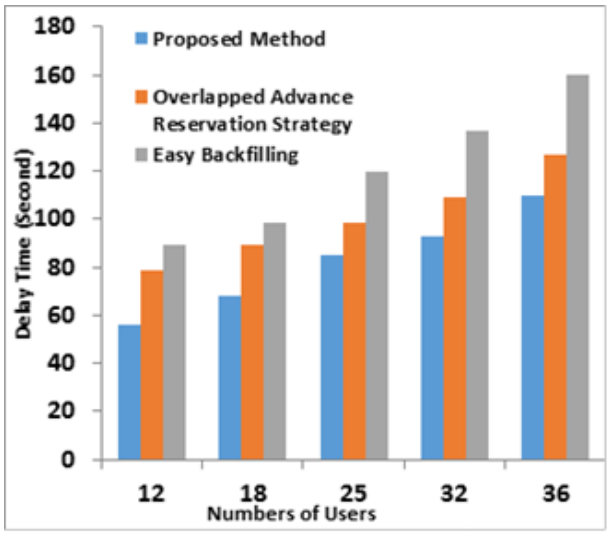


Fig 3. Effect of numbers of users on delay time

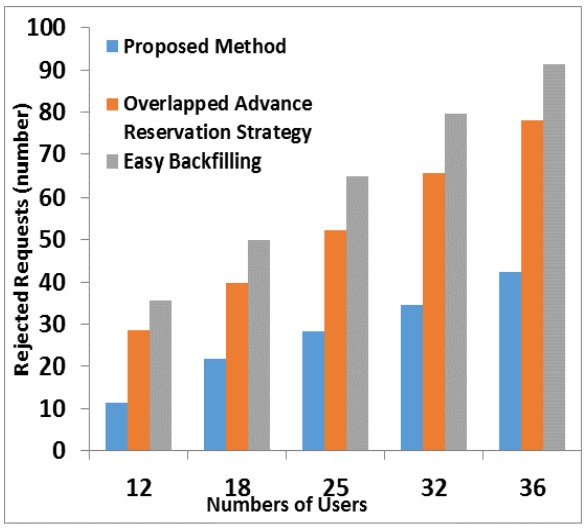


Fig 4. Effect of number of users on the number of rejected requests

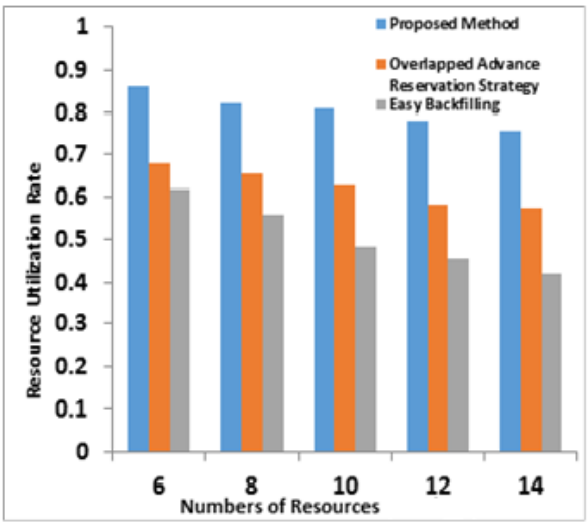


Fig 7. Effect of number of resources on resource utilization rate

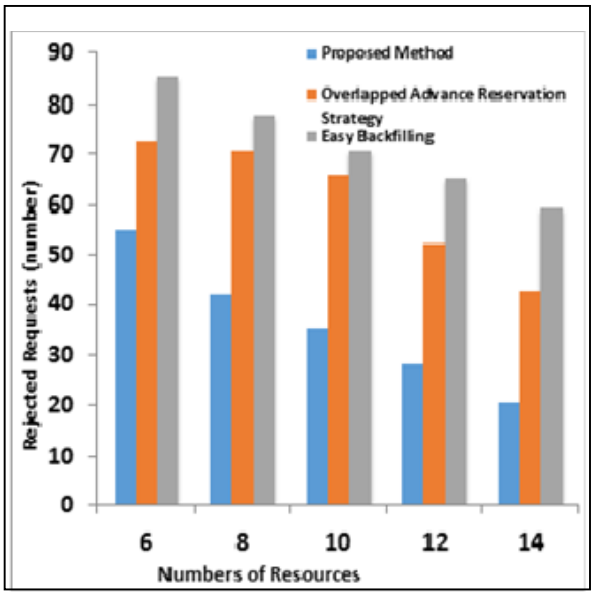


Fig 5. Effect of number of resources on the number of rejected requests

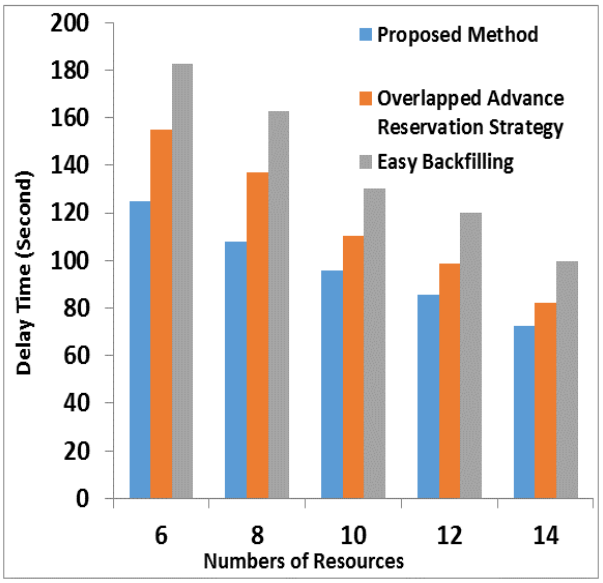


Fig 8. Effect of number of resources on delay time

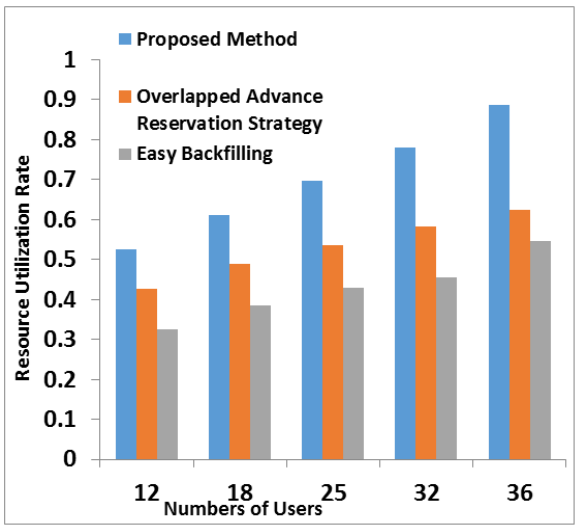


Fig 6. Effect of number of users on resource utilization rate

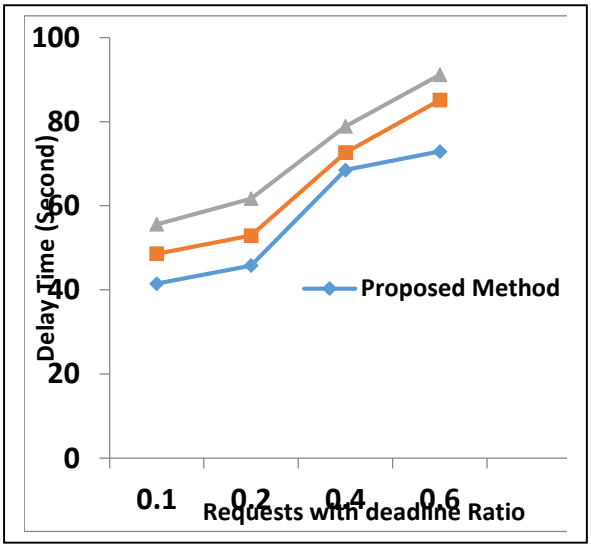


Fig 9. Effect of deadline on delay time

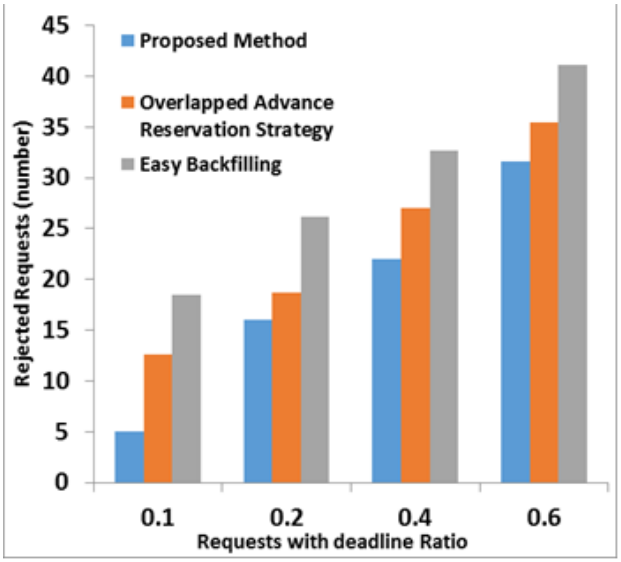


Fig 10. Effect of deadline on the number of rejected requests

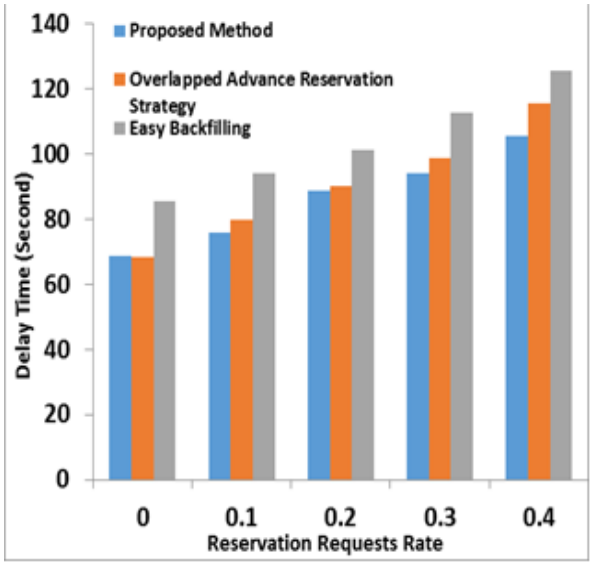


Fig13. Effect of reservation on delay time

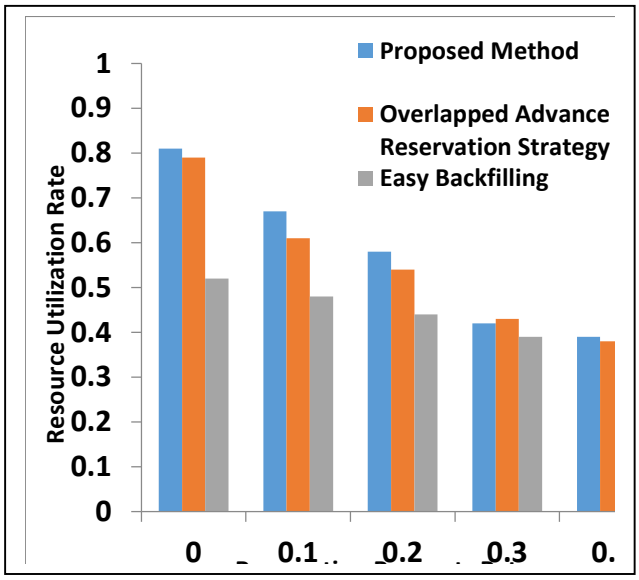


Fig 11. Effect of reservation on resource utilization rate

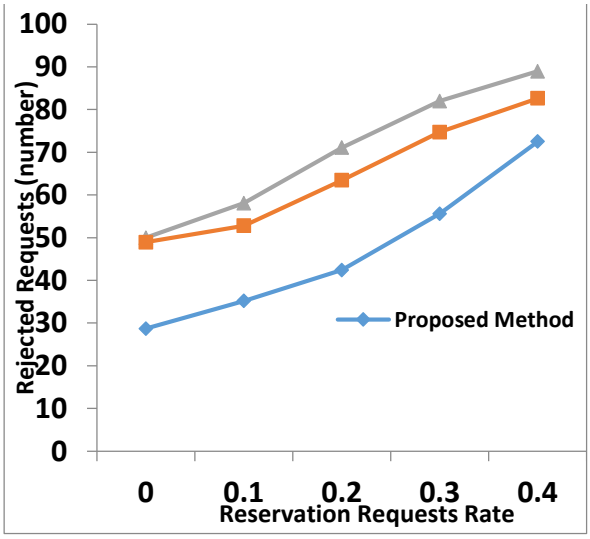


Fig 12. Effect of reservation on the number of rejected requests

REFERENCES

[1] G. F. Pfister, In Search of Clusters, 2nd Edition. Prentice Hall, 1998.

[2] I. Foster and C. Kesselman, Eds., The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2004.

[3] J. Joseph and C. Fellenstein, Grid Computing, Prentice Hall, 2003.

[4] A. Mamat, Y. Lu, J. Deogun, and S. Goddard, "Real-time divisible load scheduling with advance reservation," Proceedings of Euromicro Conference on Real-Time Systems, pp. 37-46, Jul. 2008.

[5] L. Kunrath, C. B. Westphall, and F. L. Koch, "Towards advance reservation in large-scale grids," Proceedings of Third International Conference on Systems, pp.247-252,Apr. 2008.

[6] X. Lin, Y. Lu, J. Deogun, and S. Goddard, "Real-time divisible load scheduling with different processor available times," Proceedings of International Conference on Parallel Processing, p. 20,Sep. 2007.

[7] M. Li and M. Baker, The Grid: Core Technologies, John Wiley & Sons, pp. 243-300, Apr. 2005.

[8] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," Int. J. High Performance Computing App., vol. 15, pp. 200-222, Aug. 2001.

[9] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," Software- Practice & Experience, vol. 32, pp. 135-164, Feb. 2002.

[10] W. Smith, I. Foster, and V. Taylor, "Scheduling with advanced reservations," Proceedings of the 14th International Parallel and Distributed Processing Symposium, pp. 127-132, May 2000.

[11] D. G. Feitelson and L. Rudolph, "Parallel job scheduling: Issues and approaches," Proceedings of the 1st Workshop on Job Scheduling Strategies for Parallel Processing, pp. 1-18, 1995.

[12] E. Khajevand and M. R. Meybodi, "Cost optimization in economic grid using cellular automata," Proceedings of 3rd Joint Congress on Fuzzy and Intelligent Systems, pp. 1-7, Jul. 2009.

[13] K. Khajevand and M. R. Meybodi, "Time optimization for task scheduling in economical grid computing using cellular automata," Proceedings of 3rd Iranian Data Mining Conference, pp. 1-7, Dec. 2009.

[14] W. C. Yeh and S. C. Wei, "Economic-based resource allocation for reliable Grid-computing service based on grid bank," Future Generation Computer Systems, vol. 28, pp. 989-1002, Jul. 2012.

[15] T. Ghafarian, H. Deldari, and M. R. Akbarzadeh-T, "Multiprocessor scheduling with evolving cellular automata based on genetic algorithm," Proceedings of 14th International CSI Computer Conference, pp. 431-436, Oct. 2009.

[16] M. Shojafar, S. Barzegar, and M. Meybodi, "Time optimization in economical grid using adaptive stochastic petri net based on learning automata," Proceedings of International Conference on Grid Computing and Applications, pp. 67-74, 2011.

[17] Z. Huang, P. Xiao, and D. Liu, "An overlapped advance reservation strategy for grid resources," Journal of Information & Computational Science, vol.9, pp. 2211-2220, Aug. 2012.

[18] S. Yi, Z. Wang, S. Ma, Z. Che, Y. Huang, and X. Chen, "An effective algorithm of jobs scheduling in clusters," Journal of Computational Information Systems, vol. 6, pp. 3163-3171, Oct. 2010.

[19] S. A. Makhlouf and B. Yagoubi, "Co-allocation in grid computing using resources offers and advance reservation planning," Proceedings of 8th IEEE International Conference on Advanced Computing and Communications, pp. 45-52, Dec. 2009.

[20] C. Castillo, G. Rouskas, and K. Harfoush, "On the design of online scheduling algorithms for advance reservations and QoS in grids," Proceedings of the IEEE International Parallel & Distributed Processing Symposium, pp.1-10, Mar. 2007.

[21] M. Esnaashari and M. R. Meybodi, "Irregular cellular learning automata", IEEE Transactions on Cybernetics, vol. 45, pp. 1622-1632, Aug. 2015.

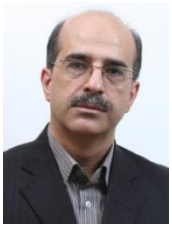
[22] N. Ganguly, B. K. Sikdar, A. Deutsch, G. Canright, and P. P. Chaudhuri, "A survey on cellular automata," Future and Emerging Technologies Unit of the European Commission, pp. 62-68, Dec. 2003.

[23] M. Abdolzade and H. Rashidi, "A cellular learning automata (CLA) approach to job shop scheduling problem," Proceedings of European Symposium on Computer Modeling and Simulation, pp. 49-54, Nov. 2009.

[24] Q. Snell, M. Clement, D. Jackson, and C. Gregory, "The performance impact of advance reservation meta-scheduling," Lecture Notes in Computer Science, Vol. 1911, pp. 137-153, 2000.

Sara Rezaei received her B.Sc. degree in Software Engineering from Parand Branch Islamic Azad University, Tehran, Iran in 2010. She received her M.Sc. degree in Software Engineering from South of Tehran Branch Islamic Azad University, Tehran, Iran, in 2015. Her research interests include grid computing, machine learning, distributed computing, and job scheduling.

Ahmad Khadem-Zadeh was born in Mashhad, Iran, in 1943. He received the B.Sc. degree in applied physics from Ferdowsi University, Mashhad, Iran, in 1969 and his M.Sc. and Ph.D. degrees respectively in Digital Communications and Information Theory & Error Control Coding from the University of Kent, Canterbury, UK.He is currently the Head of Education, National and International Scientific Cooperation Department Affairs and in the meantime the head of Post Graduate Department at ICT Research Institute (ITRC). He was the head of Test Engineering Group and the director of Computer and Communication Department at ITRC. Dr.khademzadeh is the chair of IEEE Iran Section Conferences Committee and also a lecturer at Tehran Universities.He is a committee member of the Iranian Electrical Engineering Conference Permanent Committee. Dr. Khadem-Zadeh has been received four distinguished national and international awards including Kharazmi International Award, and has been selected as the National outstanding researcher of the Iran Ministry of Information and Communication Technology.



Mansour Sheikhan is currently an Associate Professor in Electrical Engineering Department of Islamic Azad University-South Tehran Branch. His research interests include speech signal processing, neural networks, and intelligent systems. He has published more than 80 journal papers, about 70 conference papers, four books in Farsi, and six book chapters for IET and Taylor & Francis.

This Page intentionally left blank.