

Evaluation of Fault Tolerance in Cloud Computing using Colored Petri Nets

Mehdi Effatparvar

Sama Technical and Vocational Training College
Islamic Azad University, Ardabil Branch Ardabil, Iran

Seyedeh Solmaz Madani

Department of Computer and Informatics Engineering
Islamic Azad University, Ardabil, Iran

Abstract—Nowadays, the necessity of rendering reliable services to the customers in business markets is assumed as a crucial matter for the service providers, and the importance of this subject in many fields is undeniable. Design of systems with high complexity and existence of different resources in network cloud leads the service providers to intend to provide the best services to their customers. One of the important challenges for service providers is fault tolerance and reliability and different techniques and methods have been presented for solving this challenge so far.

The method presented in this paper analyzes the fault tolerance process in interconnected network cloud in order to avoid problems and irreparable damages before implementation. In the offered method, the fault tolerance was evaluated aiding colored petri nets using Byzantine technique. Summary of results analyzed by cpntools and demonstrated reliability. It was concluded that upon increase of requests, the fault tolerance is reduced and consequently reliability is also reduced and vice versa. In other word, resources management is under impact of requested services.

Keywords—Cloud Computing; Fault Tolerance; Colored Petri Nets; Reliability

I. INTRODUCTION

Upon increasing development of information technology, a great volume of computations was created that its implementation by supercomputers was very costly and not available for all. Therefore, a new technology in the name of cloud computing was emerged. Using this technology, all systems existing in a network are assumed as a computational resource and may be used for computation. Accordingly, a great and powerful source is created using the network-connected systems resources that are able to perform complex and great operation.

The cloud computations provided the requirements for using computing resources shared by the computers in different networks that are different from each other in structure as well as geographically may be situated within different intervals.

Currently, various definitions of cloud computing have been presented. The definition used in this paper is as follows:

The cloud computing is a computational model therein lots of systems are connected to each other as private or public networks to provide dynamic and scalable infrastructure for the applied programs, data storage and files. Upon emergence of this technology, the computations cost, applied programs hosting, content storage, and delivery of services was reduced

considerably. The idea of cloud computing principally is formed based on “reuse of technological capabilities” [1].

One of the important problems in cloud networks is management of time and resources, because as the viewpoint of different users, one of the most important criteria in selection of resources is implementation time. Whatever the implementation time is shorter and has lower cost, will be more appropriate as the viewpoint of users.

Cloud computing customers don't need to pay any cost for management, commissioning, maintenance and increasing the scale of their service for traffic control. They only ought to invest their cost and time for web development and pay easily the needed resources they want to provide for the web.

The cloud computing has frequent advantages, but also there are a few reasons for caution; risks such as losing services in case of occurring any problem or failure of cloud computing provider services or closure of their business. The legal problems are created when personal information has been stored in international level and security concerns are also created when the users have lost the control on their data protection. Hence, unilateral services are provided to the users for compensating the probable loss at the time of occurrence of a disaster. Fault tolerance is one of major concerns in applied programs implementation. In order to implement the applied programs correctly and reduce the effect of error on them, at first the error ought to be predicted, later managed and controlled. The fault tolerance methods are offered for prediction of these errors and performance of a proper action before fault.

The fault tolerance is the ability of a system for continuing the fulfillment of supposed tasks even in case of error. In fact, in this paper, byzantine fault tolerance technique in interconnected network cloud was used to avoid the irreparable loss, and colored petri nets were used to show the reliability of this technique.

In continue, the paper structure is as follows: in second chapter (2), the fundamental concepts related to cloud computing and fault tolerance, colored petri nets and reliability related to the subject are explained. In third chapter (3), a useful selection of previous studies and researches and similar to the subject of paper are explained briefly and usefully, and in fourth chapter (4), the proposed method is described. Finally, to show the accuracy of proposed method, in fifth chapter (5), a case study is provided for evaluation and modeling of proposed method.

II. FUNDAMENTAL CONCEPTS

Currently, different definitions of concepts are provided based on their application and performance. In this paper, the concepts were defined based on the application and type of usage. The cloud computing should not be mistaken with network computations, because in network computations, the options and information are in general provided only on the servers of a specific company, whilst the cloud computations are much more bigger and include several companies and a large number of servers and equipment.

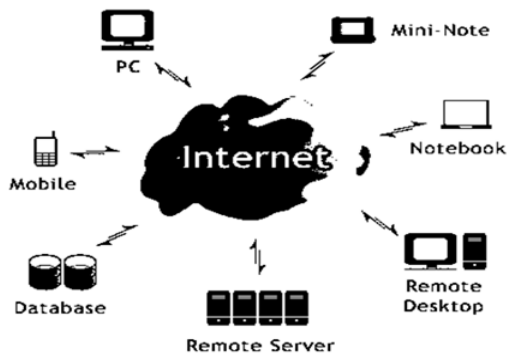


Fig. 1. Cloud computing model

A. Fault tolerance

Whereas in cloud networks one of the important topics is fault tolerance, in other word, it is necessary for the service receivers to have the reliability and security of their stored data in the highest level, therefore the service providers to retain their customers and render appropriate services ought to use specific mechanisms for applying security and high reliability factor.

The purpose of fault tolerance in cloud systems is that in case of occurrence of error, the system to have the ability of tolerating to the happened occurrence and can continue its process. In this type of systems, definitions of error, fault and failure are presented to associate the difference between them in the mind of reader. Therefore:

- Failure: whenever a system doesn't perform its expected job correctly, a failure has been occurred.
- Fault: The reason of failure occurrence is existence of an error in the system.
- Error: The reason of fault is existence of an error in the system.

B. Colored petri net

These nets present a graphic and clear exhibition of system together with a mathematical approach and can show the communication patterns, control patterns and information processes. These nets provide a framework for analysis, validation and evaluation of performance, the basis of petri nets has been formed based on the graph and informally it is a two-part directional graph consisted of two elements of time and transition. These nets are status-based and not event-based and it makes the explicit status modeling of each case possible.

Petri nets provide models of structural and behavioral aspects of a discrete event system. Moreover, provide a framework for analysis, performance validation and evaluation, and reliability [2].

Colored petri nets figure 2, provide exacter models of complex asynchronous processing systems. In these nets, contrary to the petri nets, the tokens are distinguishable from each other, because each one of tokens has traits in the name of color. This type of nets provides exacter and detailer modeling from complex asynchronous processes. The tokens may be different from each other, so that a property called color is added to each token. The arcs may include mathematical phrases consisting of combination of color sets and variables related to them. Guard is a Boolean expression that is attributed to a transition and creates conditions for activation of input arc. In colored petri nets, each one of places, arcs and transitions depending on their color have their own guard [2].

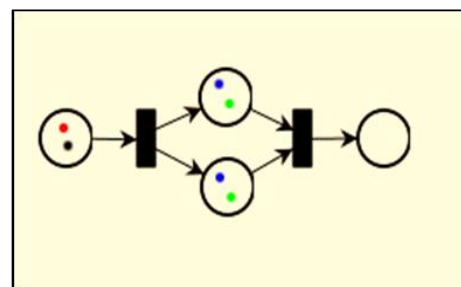


Fig. 2. A model of colored petri net

C. Reliability

However, nowadays the cloud nodes have been considered by the people so much, nonetheless one of the major challenges that cloud computing face it, is procedure of data protection and applying security for users processes. The security that is provided in the cloud environment is very important for the organizations and the people, because a few organizations suppose the transfer of important applied programs and their sensitive data to a general cloud environment as a great risk. Therefore, to reduce these concerns, a cloud provider must make this confidence that the customers can reserve their security and privacy control on the applied programs, so the cloud providers to convince their customers about the security issues ought to perform actions such as service level agreement. This agreement is a document that specifies the relationship between provider and receiver and indeed is a legal agreement between service provider and customer.

Following items are raised in the agreement about satisfying and ensuring the customer [3]:

- Identification and definition of customer needs
- Simplification of complex problems
- Reduction of grounds for conflict between users
- Persuading to discourse about encounters and disputes
- Omission of unreal expectations
- Presentation of a framework for easier perception

Identify applicable sponsor/s here. If no sponsors, delete this text box (sponsors).

The cloud computing has not always provided continuous reliability.

III. PREVIOUS WORKS

Within recent years, various methods were propounded for evaluation of fault tolerance and reliability in cloud networks and these methods were studied in various researches. A few cases are analyzed in this paper.

Zh.Shan simulated and analyzed the performance in grid system in consideration of priority queues and applying priority for tasks and subtasks. This model has been made based on stochastic petri nets and analysis of its performance was provided based on properties of petri nets [4].

In another model, the modeling and simulation of tasks scheduling procedure in grind environment was analyzed and studied to reach the maximum operational power and reliability. This study is based on the queue systems and has no formal definition and ultimately the simulation was made based on the petri nets [5].

Genetic algorithm that is a hybrid evolutionary algorithm was presented for solving independent tasks scheduling problem in cloud network. The main objective in this algorithm is finding a solution therein the overall implementation time is minimum. Whereas genetic algorithm searches throughout the problem environment and is weak in local search, upon its combination to thermal simulation that is a local search algorithm, it is attempted to remove this defect and so the combination of advantages of these two algorithms were used. The chromosomes are exhibited by RFOH algorithm. Genetic algorithm includes a random population producer, elitist selection operator, repetitive combination and mutation aiding thermal simulator. Based on the fitness function, selection operator selects half of the best chromosomes from population. The combination operator is running and so the new children are generated for the next generation [6].

In a method, to evaluate the reliability of byzantine fault in a system, byzantine fault tolerance technique was used. In this method, interconnected network cloud was used. This method includes the characteristics such as an automatic job scheduling instrument that allows the job plan to be provided automatically to several heterogeneous clouds, and a message system that creates the secure connection between the cloud and interconnected networks cloud and a fault tolerance adjudication system was presented as well. In this method, to prove the accuracy of method, two virtual machines were tested experimentally [7].

A method was presented for tasks distribution modeling and reliability calculation in cloud networks with star topology therein the tasks scheduling for reaching to the appropriate quality level as one of important and outstanding fields in cloud networks was analyzed. In this method, the reliability in grid services was examined and using the colored petri nets, a model was offered for computation. In this study, the grid environment has star topology and consequently RMS is connected with all resources in the grid. As specified, task of RMS is receiving tasks consisted of a series of subtasks from user and later distribution of the corresponding subtasks on the resources available on the grid. The general schema of job of

tasks receipt and division thereof to subtasks by RMS and distribution among the resources is shown in fig. [3].

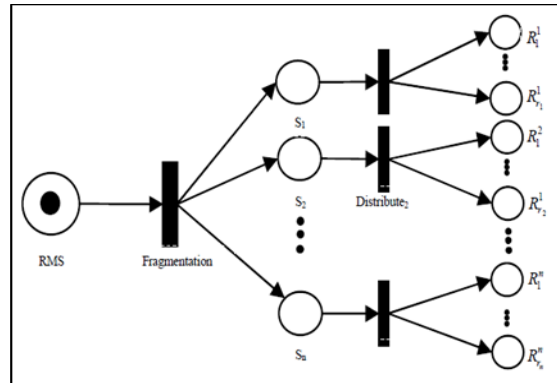


Fig. 3. Breaking the tasks and distribution of subtasks by RMS [3]

RFOH algorithm was presented for task scheduling with fault tolerance in computational grid aiding colored petri nets. This method records the history of fault event in resources in a table called fault event history table in information server of grid. Each row of FOHT table for each resource includes two columns; one column shows the history of failure event in that resource and the other one specifies the number of tasks implemented by that resource that was modeled by colored petri nets [8].

The modeling along with reliability computation methods with lower performance was studied and only a few case and specific studies were applied on the grid. In a few other methods that Y-Sh.Dai et al computed reliability in grid services, it was concluded that these studies provide the computations related to reliability and performance in grid services and only a solution for their calculation and maximization was presented and no virtual model thereof was presented [9].

In another method, the workflow procedure in grid environment was examined based on service and using UML charts [10]. In another method, workflow modeling in grid environment was discussed using simple petri nets and FhRG software. In this paper, the resulted petri network was presented, but this network was simple and allocated for the said software [11].

The hybrid HPSO algorithm in fact is combination of PSO and thermal simulation that upon using thermal simulation gets away from falling into a local optimum. To map the scheduling problem to a solution in this paper, the research environment is assumed as $N \times M$ dimensions, therein N is referred to the number of subtasks and M to the number of resources. Each particle is comprised of M parts and each part has N independent tasks that show the combination of N tasks on M machines [12].

A new scheduling algorithm was designed based on two basic Min-Min and max-Min algorithms so that the advantages of these two algorithms were used and yet the faults were covered. The criterion for selecting the proposed algorithm, the algorithm selected from two foregoing algorithms, is standard deviation of tasks completion period on resources. Min-Min

algorithm includes two stages; at the first stage, the waiting time for each subtask is specified and in the next stage, the tasks are sorted and marked based on job completion time on descending basis.

The tasks are allocated to their corresponding resources based on the priority of resources and this process is continued until all duties existing on MT are processed. Max-Min algorithm is similar to Min-Min algorithm with this difference that at the first stage, the tasks and jobs are sorted and marked based on the job completion time on ascending basis [13].

In another method, genetic algorithm was presented for solving the dependent tasks scheduling therein two important parameters of service quality including time and cost were taken into account. In this algorithm, instead of production of initial population randomly, disturbed variables were used. The combination of genetic algorithm advantages to the disturbed variables resulted in distribution of produced solutions by this algorithm throughout the research space and avoided the early convergence in the algorithm and the better solutions and products to be achieved within shorter time, and the algorithm convergence speed to be increased [14].

In a method, an algorithm was presented using queue theory for reduction of programs running cost in cloud network environment. The algorithm presented in this method is system-oriented and in addition to considering the performance and productivity factors focused on cost parameter that is raised more in business cloud environments [15].

IV. PROPOSED METHOD

The proposed method is formed based on byzantine fault tolerance. The interconnected clouds, in other word multiple clouds that each one has a specific policy and management and also is administrated differently were studied and analyzed using this technique. Colored petri nets were used to show the fault tolerance in cloud network aiding this technique. In addition, to avoid extra costs before implementation and execution phase, the respective method was simulated by colored petri nets aiding cpntools to evaluate the reliability in the proposed method.

Byzantine fault tolerance is not used for single cloud networks considering reducing the network reliability [7].

In continue, to introduce the method, at first byzantine tolerance and thereafter reliability computation procedure in the proposed method was analyzed and ultimately upon evaluation of fault tolerance and computation of reliability aiding colored petri nets and presentation of an executive model of proposed method in cpntools, obtained results were computed and exhibited.

A. Byzantine fault tolerance technique in interconnected cloud computing

Whereas cloud networks are assumed as one of main principles of computational systems, reliability in these systems is one of essential concerns, therefore the error potential in this type of systems is due to high simplicity. Hence, different techniques were developed for fault tolerance in these systems and in this paper byzantine technique was used.

In fact, byzantine fault is appeared when therein the server may provide any response optionally simultaneous to emergence of problems such as enemy attacks, user fault and software fault. A solution for avoiding this problem is use of byzantine fault tolerance (BFT). Byzantine fault tolerance figure 4, is the name of a problem that is occurred in distributed systems that is occurred due to creation of intentional failures.

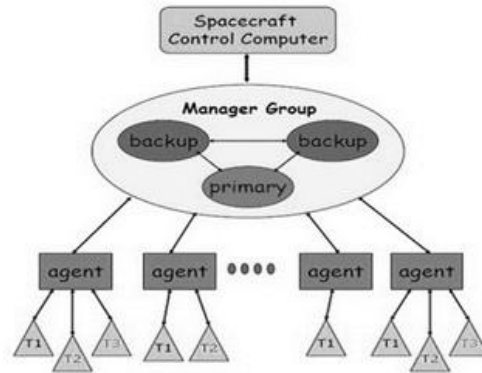


Fig. 4. Byzantine fault tolerance

FT-FC (Fault Tolerant – Fault Control) is one of fault tolerance frameworks in interconnected network cloud that that has various features such as having an automatic job scheduling tool that provides the possibility of showing job plan automatically to several heterogeneous clouds and a message-based system that makes the safe communication between the cloud and FT-FC and also includes a fault tolerance adjudication system. Figure 5, exhibits the system created based on FT-FC framework. The modules shown in the figure are described in continue.

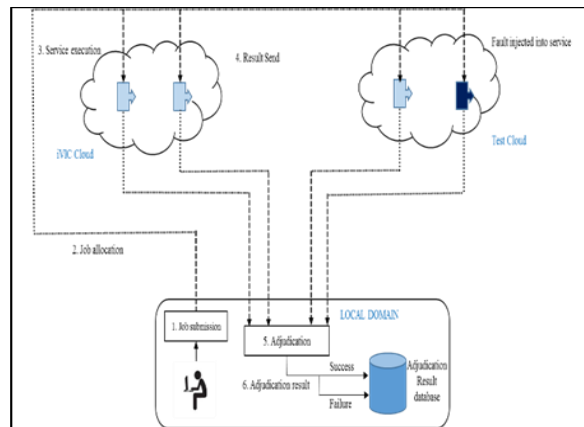


Fig. 5. System created based on FT-FC framework

In figure 5, it is shown for two clouds titled Test, iVIC with FT-FC framework. The purpose is expression of clouds independent from each other. The used modules respectively are as follows:

- Job submission: Division of jobs automatically among distinct clouds.
- Job allocation: The jobs are allocated to N-Copy services.

- Service execution: Each cloud has a backup that is switched thereto in case of occurring any error.
- Result send: Sending the results to adjudication node by services.
- Adjudication: Decision making of adjudication node in time management and accuracy of the result of executive process.
- Adjudication result: Specifying the accuracy and inaccuracy of obtained results.

The obtained results are stored in local database. In consideration of the foregoing, at the next stage, the reliability evaluation in this method should be assessed and in continue its procedure is described.

B. Reliability computation procedure

In this study, to compute the reliability, a failure rate is labeled to each one of nodes. However, all of these rates are designed dynamically in the model so that to be closer to the reality. To assess the reliability, equation (1) is used [16].

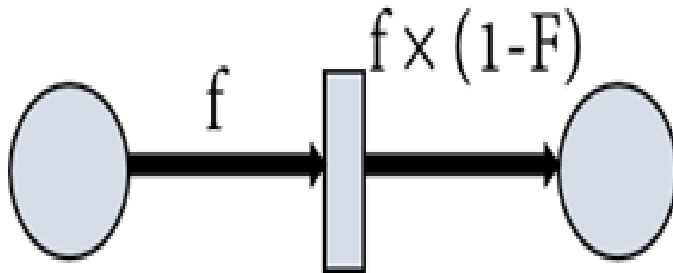


Fig. 6. equation (1) used to assess the reliability

In above equation, f denotes reliability and F the error rate or failure factor. After that each time in case of error occurrence, the reliability was computed, is averaged and updated with the new event of failure and new reliability.

In other word, in each cloud, an error factor or fault may be occurred due to byzantine error reasons. Equation (1) is used for evaluation of fault tolerance and a backup is used in each cloud for upraising the fault tolerance threshold limit of byzantine framework.

C. Creation of executable model

In this paper, colored petri nets and CPN Tools were used for creation of executable model. Reliability is evaluated in FT-FC framework using byzantine fault tolerance upon labeling the failure rate in each cloud. In continue, figure 6, shows the executable model of a distinct cloud network.

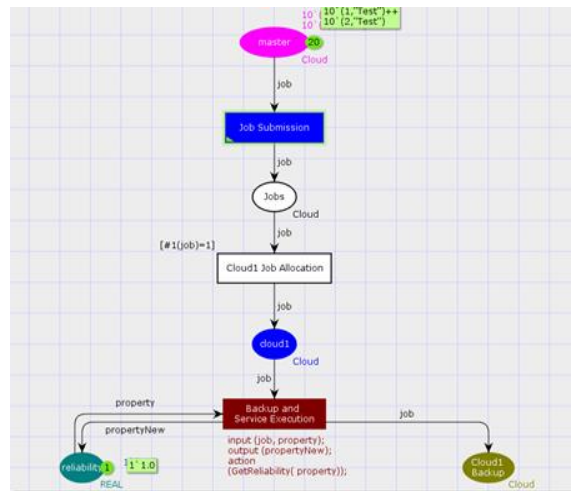


Fig. 7. Executable model of a distinct cloud network

V. CASE STUDY

In this paper, to show the applicability of method and accuracy of offered method, an example of cloud network with five distinct clouds were analyzed due to high complexity. Metric analysis of reliability is provided using the proposed method and simulation of executable model.

In order to evaluate the accuracy of performed simulation, Master provides five commands for sending to the clouds that each one of these commands were specified in the name of Job Properties. To show the dynamism in job, each command is repeated 10 times and 30 times; in other word, the possibility of data exchange between Master and clouds was shown.

figure 8, is a general schema of presented model that was simulated based on the explained assumptions and obtained results are analyzed in continue.

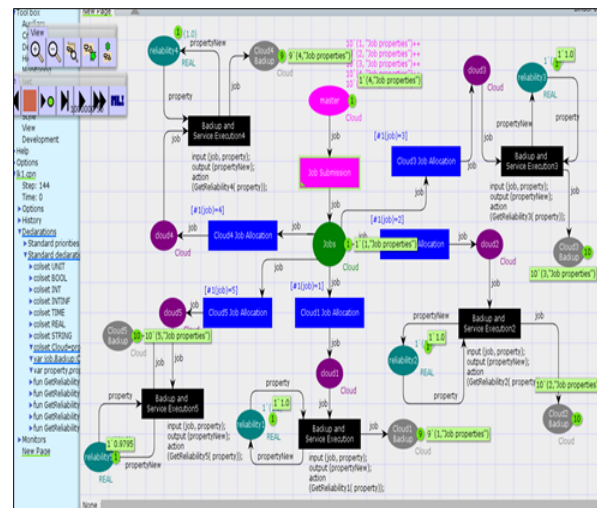


Fig. 8. General schema of presented model

In table 1, summary of results of execution of number of different jobs on the presented method is shown by byzantine fault tolerance technique. The failure rate in model is dynamic.

TABLE I. SUMMARY OF RESULTS OF PROPOSED MODEL

Id	Job Submission	Fault rate	Reliability	Average Reliability
cloud1	10	0.21	0.895	0.954
cloud2	10	0.11	0.945	
cloud3	10	0.13	1	
cloud4	10	0.09	1	
cloud5	10	0.14	0.93	
cloud1	30	0.21	0.368474398	0.777050282
cloud2	30	0.11	0.8945	
cloud3	30	0.13	0.874225	
cloud4	30	0.09	1	
cloud5	30	0.14	0.74805201	

According to the results obtained from simulation of studied method in this paper, the effect of increase and decrease of number of jobs on fault tolerance value and finally total reliability of system is observable.

In other word, the resources management considering the increase of requests will intensify the faults and the fault tolerance is declined. As a result, reliability of system is reduced. figure 9, shows the results of reliability evaluation with 10 and 30 tasks for each job in the simulated model.

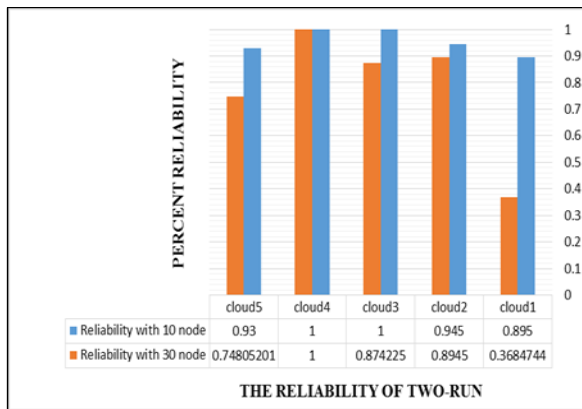


Fig. 9. Reliability evaluation by failure rate for 10 and 30 requests for each job

Byzantine fault tolerance n interconnected clouds that are stronger in management and have powerful infrastructure and are managed by the great cloud producers such as Amazon and Google is higher than byzantine fault tolerance with optional resources clouds that include lots of users and combined their computational resources.

In optional resources, due to dynamism of environment, reliability of a system is a critical issue and a byzantine fault tolerance system was presented for solving this problem. If number of our resources is $3F+1$, F tolerates byzantine error. Commonly, infrastructure of clouds with optional resources is cheaper and more dynamic than greater clouds, but has lower power and reliability. Moreover, communication links between modules is not reliable [17].

In table 2, results obtained from execution of number of different works on byzantine fault tolerance method in clouds with optional resources is shown.

TABLE II. RESULTS OBTAINED FROM BYZANTINE FAULT TOLERANCE IN CLOUDS WITH OPTIONAL RESOURCES

Id	Job Submission	Fault rate	Reliability	Average Reliability
cloud1	10	0.21	0.7465553	0.85702715
cloud2	10	0.11	0.797493651	
cloud3	10	0.13	0.935	
cloud4	10	0.09	1	
cloud5	10	0.14	0.8060868	
cloud1	30	0.21	0.361424654	0.637855983
cloud2	30	0.11	0.64561702	
cloud3	30	0.13	0.576647074	
cloud4	30	0.09	1	
cloud5	30	0.14	0.605591166	

In figure 10, a comparison between proposed model and byzantine fault tolerance method is provided in clouds with optional resources.

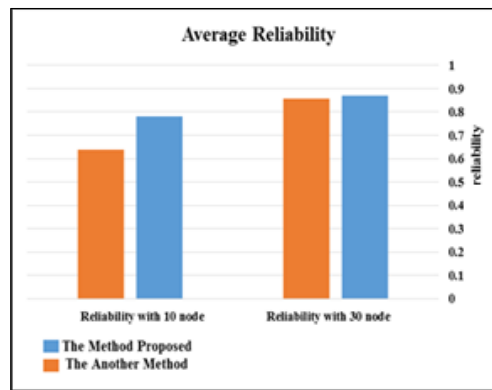


Fig. 10. Comparison between proposed model and byzantine fault tolerance method in clouds with optional resources

VI. CONCLUSION

The method presented in this paper analyzed byzantine fault tolerance in interconnected network clouds and colored petri nets were used to evaluate the reliability aiding this method that has strong math support. According to the results obtained from simulation, the fault tolerance using offered method is appropriate in comparison to the fault tolerance method in clouds with optional resources.

One of advantages of proposed method is combination of byzantine technique and colored petri nets that reliability was evaluated and analyzed using provided modeling. It was concluded that upon increase of requests, the fault tolerance is reduced and consequently reliability is also reduced and vice versa. In other word, resources management is under impact of requested services. One of limitations of proposed method for the subsequent activities is that in case the fault gets beyond the threshold limit and cloud tolerance, how reliability may change or which technique can be used to answer this question if the best option is tolerance or fault removal.

REFERENCES

- [1] Asgharpour, Mohammadjavad, Multi-criteria Decision Making, 2011, 10th Ed., Tehran, Press Institute of University of Tehran;
- [2] Kristensen L.M., Wells L. Lensen K., "Coloured Petri Nets and CPN Tools for modeling and validation of concurrent systems," International Journal on Software Tools for Technology Transfer (STTT), no. Springer Berlin / Heidelberg, pp. 213-254, 2007.

- [3] Entezari Maleki, Reza; Abdollahi Azgami, Mohammad, Subtasks Distribution Modeling and Computation of Reliability in Grid Services with Star Topology using Colored Petri Nets, March 2009, 14th Annual Conference of Iranian Computer Association;
- [4] Shan Zh., Lin Ch., Ren F., Wei Y., "Modeling and Performance Analysis of a Multiserver Multiqueue System on the Grid", Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03), 2003.
- [5] Caron E., Garonne V., Tsaregorodtsev A., "Defination, modeling and simulation of a grid computing scheduling system for high throughput computing", Future Generation Computer Systems 23, pp. 968-976, Elsevier, 2007.
- [6] Cruz-Chavez, M. , Rodriguez-Leon, A. , Avila-Melgar, E. , Juarez-Perez, F. , Cruz-Rosales, M. and Rivera-Lopez, R. "Genetic-Annealing Algorithm in Grid Envirinment for Scheduling Problems", Security-Enriched Urban Computing and Smart Grid Communications in Computer and Information Science, Springer, Vol. 78, pp. 1-9.2010.
- [7] Garraghan.p and Townend.P and J. Xu, "Byzantine FaultTolerancein Federated Cloud Computing" Proceedings of The 6th IEEE International Symposium on Service Oriented System Engineering, pp. 280-285, 2011.
- [8] Khanli, L., Etminan Far, M., Ghaffari, A., Reliable Job Scheduler using RFOH in Grid Computing, Journal of Emerging Trends in Computing and Information Sciences, Vol. 1, No. 1, pp. 43- 47, 2010.
- [9] Dai Y.S., Levitin G., "Optimal Resource Allocation for Maximizing Performance and Reliability in TreeStructured Grid Services", IEEE Transaction on Reliability, Vol. 56, No.3, September 2007.
- [10] BenDaly Hlaoui Y., Jemni BenAyed L.; "Toward an UML-based Composition of Grid Services Workflows".Research Unit of Technologies of Information and Communication, Tunisia, ACM, AUPC'08, July 2008.
- [11] Neubauer F., Hoheisel A., Geiler J., "Workflow-based Grid applications", Future Generation Computer Systems 22, pp. 6-15, Elsevier, 2006.
- [12] Chen, R., Shiau, D. and Lo, SH., Combined Discrete Particle Swarm Optimization and Simulated Annealing for Grid Computing Scheduling Problem, Lecture Notes in Computer Science, Springer, Vol. 57, pp. 242 – 251, 2009.
- [13] Etminani, Kobra, Design and Implementation of a Scheduling Algorithm in Grid based on two Min-Min and Max-Min Algorithms and Evaluation of its Performance, 2007, Thesis of Ferdowsi University of Mashhad.
- [14] Gharooni fard, G., Moein darbari, F., Deldari, H., Morvaridi, A., Scheduling of scientific workflows using a chaos- genetic algorithm, Procedia Computer Science, Elsevier, Vol. 1, No.1, pp. 1445- 1454, 2010.
- [15] Afzal, A., McGough, A.S., Darlington, J., "Capacity planning and scheduling in Grid computing environment", Journal of Future Generation Computer Systems 24 , pp.404-414, 2008.
- [16] Farjaminejad.F and Harounabadi.A , "Modeling and Evaluation of Performance and Reliabilityof Component-based Software Systems using Formal"International Journal of Computer Applications Technology and Research Volume 3– Issue 1, 73 - 78, 2014.
- [17] Zhang .Y and Z.Zheng and M. Lyu, "A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing" IEEE 4th International Conference on Cloud Computing, pp. 444-451, 2011.