International Academic Institute for Science and Technology

International Academic Journal of Science and Engineering
Vol. 2, No. 2, 2015, pp. 274-284.
ISSN 2454-3896

International Academic Journal of Science and Engineering

www.iaiest.com

Modelling the Best-client Data Replication Method based on Stochastic Reward Nets

Zahra Sotudenia[a] , Hassan Motallebi[b]

[a] *Kerman Islamic Azad University, Kerman, Iran.*

[b] *Graduate University of Advanced Technology, Kerman, Iran.*

Abstract

In order to improve the data access performance in data grid, multiple replica of each data items are created in several data sites. However, due to the storage capacity constraints of data sites, replication costs and access pattern dynamicities it is hard to decide which data item from which data site should be replicated on which data site and when. Improving performance metrics such as response time and throughout are the main concern in many research works in the area of data grids. Numerous data replication policies are proposed for data replication in data grid and in order to evaluate these policies a stochastic simulation approach is employed. In this paper, a modelling framework, based on stochastic reward nets, is designed for modelling and evaluation of data replication policies in data grid. In this framework, separate sub-models are designed for workload generation, file replacement modules and the synchronization timers. Using the proposed framework, a wide family of replication policies can be specified and evaluated. Each replication policy can be simply specified by putting together the designed sub-models. We also employ the proposed framework to specify the best-client data replication method.

Keywords: data grid, data replication, best-client policy, stochastic reward net

Introduction:

In this paper a modeling framework is proposed for specifying the data replication policies in data grid. In this framework, the behavior of the system in accessing different copies of a typical data item in different sites is modeled. In the proposed framework we design different sub-models for specifying the workload generation module, the file replacement behavior and the replication timer and etc. We show that these sub-models can be simply put together to specify replication policies such as the best client policy, cascade policy and fast spread policy. Using the resulted models the data replication policies can be compared with each other from the data access performance point of view.

The workload on the data grid benefits from various types of data locality including special, temporal and geographical locality. While considering these localities in a SRN model is quite a challenge, we consider only the temporal locality in data references. The temporal locality of reference means that the data item accessed recently is likely to be accessed in the near future. In order to take into consideration the temporal locality of reference [1], we employ workload models in which the rate of request for a data item experiences smooth changes over time. This type of behavior describes both the temporal locality and access rate fluctuations which are part of the behavior of a real life workload.

Another aspect of the behavior of the data grid is the replacement of a data with a more popular file which is frequently needed. The reason for removing files is that, for the sake of performance improvement the limited storage of the data site should be occupied by more popular files. In order to model the file replacement behavior a sub-model is designed where the rate of file replacement is determined with respect to the rank of the file. The rank of each data file is a positive integer between 0 and a maximum rank k. Once a data access request is served in a data site, the rank of the file is increased. The rank of a file may also decrease through time as the file becomes old. The removal rate of each data file is a function of the rank of that file.

In the proposed framework, we assume that the access requests are initiated from the leaf nodes and once the request is received at a node, in case a local copy of the requested file is available on the site, the request is served locally and the rank of the accessed file in increased by one otherwise the request is forwarded to another site. In other works, the internal nodes of the grid never initiate an access request.

The remainder of this paper is organized as follows. In Section 2, we give some definitions about the data grid. Section 3 presents the proposed modelling framework. First, the workload generation module, the replication timer and the file replacement module are described and then the best-client replication policy is specified based on the SRN model. Finally, Section 4 provides concluding remarks.

Data Grid

A grid is a distributed environment where heterogonous and geographically distributed resources are shared with the aim of processing large computational problems. From the users' perspective, a grid is a collaborative environment for solving resource intensive problems which provides a high level of transparency [2-6]. Data grid is a special type of grid environment with the aim of dealing with data intensive problems. A data grid provides services for storing, sharing, distributing and modifying high amounts of data resulted from scientific experiments [7-9]. There are multiple topologies for a grid

environment including tree-based topologies, ring topologies and hybrid topologies [10-12].

In order to increase availability and reliability, improving performance metrics and fault tolerance capabilities, etc. several replications of a data item are created at different data sites. However, replication brings the need for consistency among different copies of each data item. It also increases the amount of storage needed for keeping replica of the items.

Several methods are proposed in the literature for maintain consistency among replica, some of which are discussed in [13-19]. However, in this paper we assume that all the access requests are read requests which does not need the consistency concerns. We also assume that the underlying grid environment has a tree-based topology [15, 20-22]. We propose a framework for specifying and evaluating replication policies. The kind of evaluation considered in this study is the mean value analysis where the average behavior of a systems is studied.

The proposed framework

In this section, we describe the proposed modelling framework. First, we give the challenges in modelling data replication policies. Then, we specify the workload generation module, the replication timer and the file replacement module. Finally, we model different replication policies based on the SRN model.

3.1 Challenges

There are some challenges in petri net based specification of replication policies some of which are described in the following:

- **Modeling reference locality**. The proposed model for describing the sequence of generated access requests should consider the locality of references.
- **Modeling access rate fluctuations**. In a real scenario, the rate of request arrivals experience increases and decrease durations. Therefore, there are fluctuations in the access rate which should be considered in the modeling framework.
- **Modeling the file removal behavior.** In each grid site several data files may be kept, due to the storage constraints, some old and non-popular files should be removed when new popular ones arrive. The proposed framework should have a strategy of ranking files according to their instantaneous popularity which lets us decide which file to remove when a new file is brought to the site.
- **Modeling the file ranking schemes.** Different data replication policies use different schemes for file ranking. These file ranking schemes should be modeled in the proposed framework.
- **Modeling non-exponential activities in SRN models.** Many of the activities in the data grid environments cannot be approximated by the exponential timed transitions. Modeling these activities using the SRN model is a challenging issue.
- **Modeling different data replication policies.** Different aspects of the data replication policies should be considered in the proposed framework.

The modeling framework proposed in this paper addresses these challenges. In order for modeling reference locality and modeling access rate fluctuations we use some workload generation models in

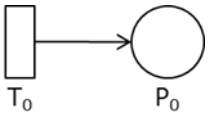
which both issues are considered.

3.2 Workload generation

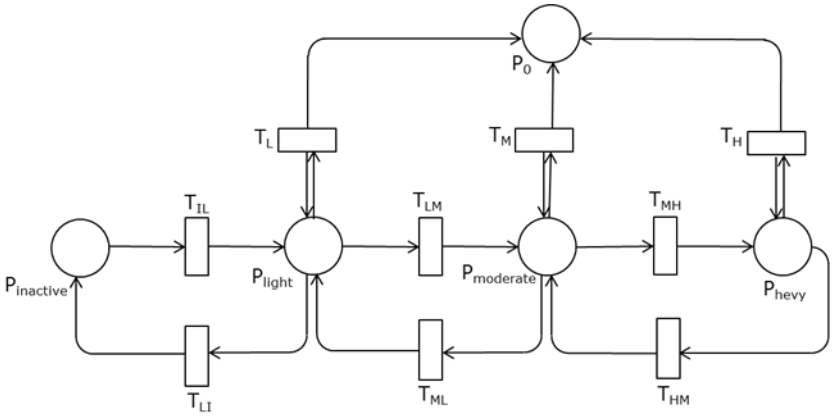
The workload model specifies the process of access request arrivals. We use some workload models that consider the fluctuations in the request arrival rates. It's worth mentioning that in real world the workloads may experience sudden changes in their intensity and may have quite complicated patterns. However, since real workloads are too complicated to be analyzed, we use the following workload models that are both realistic and analyzable. In these workload models, we approximate the actual workload model with an analyzable model which is both easy to analyzable and accurate enough. We use the following workload models:

- **The Poisson workload model.** The poison workload model assumes that the request rate is constant through time [23, 24]. The SRN model for this model is shown in Figure 1(a). The place P_0 in this model represents the queue of arrived requests and the execution of the transition T_0 denotes the arrival of a new request with the rate λ.
- **The $MMPP_1$ workload model.** This workload model is based on a markov modulated poison process (MMPP). In fact it is a Poisson process with the difference that the request rate is not constant and may change in different state of a modulating markov model. The instantaneous rate of the process is determined with respect to the current state of a markov model. The SRN model for this workload model is shown in Figure 1(b). In this model, we have four levels of workload intensity, represented by the places $P_{inactive}$, P_{light}, $P_{moderate}$ and P_{heavy}.
- **The $MMPP_2$ workload model.** This workload model is also based on a MMPP. The SRN model for this workload model is shown in Figure 1(c). In this model, the marking of the place P, represents the load intensity. By firing the timed transition T_0 (T_1) by the rate λ (μ) the marking of the place P or the load intensity may increase (decrease). The timed transition T_3 with a marking dependent firing rate $\#P \times \sigma$ represents the arrival of access requests. The arrival rate is determined by multiplying that constant σ by the number of token in place P. The rightmost part of the model in Figure 1(c) is the same as the model in Figure 1(a) with the only difference that the firing rate of the transition T_3 is controlled by the number of token in place P.

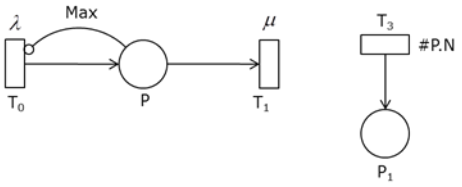
In the above workload models, the only part of the model that communicates with other parts of the data replication model is the place containing the arrived requests.



(a)



(b)



(c)

Figure 1. The proposed workload models (a) Poisson, (b) $MMPP_1$ and (c) $MMPP_2$

3.3 Replication Timer

In some of the replication policies, the process of replicating popular data items is performed every T time units. Therefore, we need a timer that ticks every T time unis and reminds us to replicate the required

files. In Figure 2, the SRN model of our replication timer is shown. This timer ticks every T time units and adds a token to the place P_1.

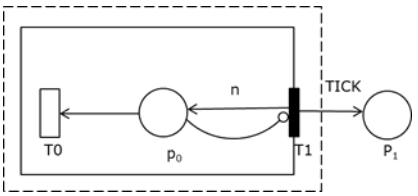


Figure 2. The SRN model of the replication timer

As the transition T_0 fires, the number of token in P_0 decreases until it becomes empty. Once the place P_0 becomes empty, the immediate transition T_1 fires and n token are added to the place P_0. Since the firing rate of the transition T_0 is m, on average every $\frac{1}{m}$ time unit T_0 fires and it takes $\frac{n}{m}$ time units for the place P_0 to become empty. Therefore, every $\frac{n}{m}$ time units, the immediate transition T_1 fires and the replication timer ticks. Thus, the values of the two parameters n and m should be chosen such that n/m equals T. As P_0 become empty the transition T_1 fires as it fires a token in added to the place P_1 and n tokens are added to the place P_0. This simulates the behavior of an hourglass in the next round.

3.4 File replacement module

Since due to storage capacity constraints in data sites, only a limited number of files can be stored in a grid site, we need a method for deciding which file should be kept and which one is better to be replaced. In order to make appropriate decisions we need to define a rank metric for each file. The rank metric specifies how popular the file is. As the rank of a file decreases the file is more likely to be replaced with a newer file. Arriving access requests for a file should increase the rank and time passing should decrease it. In the SRN model in Figure 3, the behavior of our file replacement module is specified.

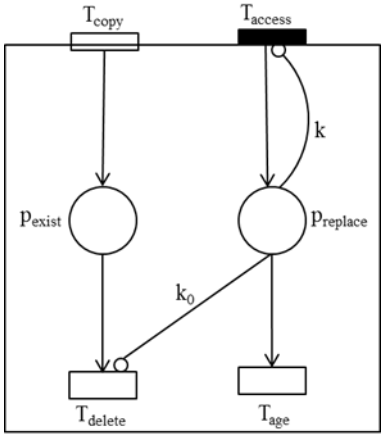


Figure 3. The SRN model of the file replacement module

In this model, the rank of file is represented by the number of tokens in place $P_{replace}$. The file rank is a number between 0 and k. With each access to the file and firing of the transition T_{access}, the rank of the file is increased. As time goes by, the transition T_{age} fires and the rank of the file is decreased. Choosing the value for the firing rate of the transition T_{age} is very important and can be obtained experimentally. The existence a token in place P_{exist} means that a local copy of the file exists. In case the local copy does not exist, the arrived access requests for the file should be forwarded to the parent node. Since due to the storage capacity constraints, only a limited number of files can be kept in a data site, the limited storage of the data site should be allocated to the most popular files. Therefore, once the rank of a files becomes zero, the file may be removed with the rate δ. The file removal is represented by the timed transition T_{delete}.

3.5 Specifying the best client policy

In the best client policy, each node maintains some state variables about the number of request arrived form clients and the identity of the clients. These variables are reset at the start of each time interval. At the end of each time interval, the values of these variables are examined to see if the number of request for a file has reached above the threshold. In case a file has become popular, a copy of the file is sent to client with most requests (best client). Also, all state variables for that file are set to zero. Although the number of the leaves in this model and other models is assumed to be four, the method and the models are extensible for the general case. The SRN model of the behavior of the leaf nodes in the best client policy is shown in Figure 7. The model is very similar to its counterpart in the cascading policy, with the

difference that in case the local copy is not available, the request is forwarded to the root node. The SRN model of the behavior of the root node is discussed in the following.

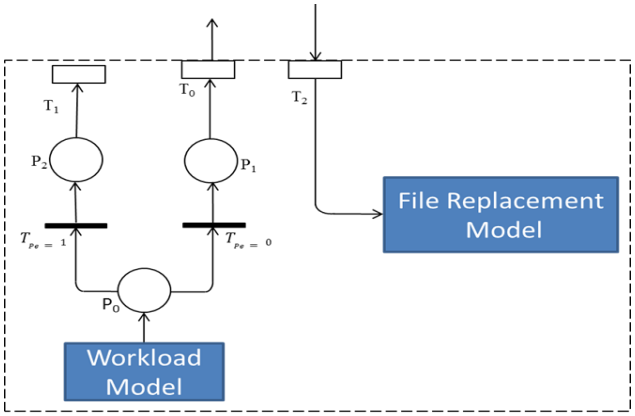


Figure 7. The SRN model of the leaf nodes in the best client policy

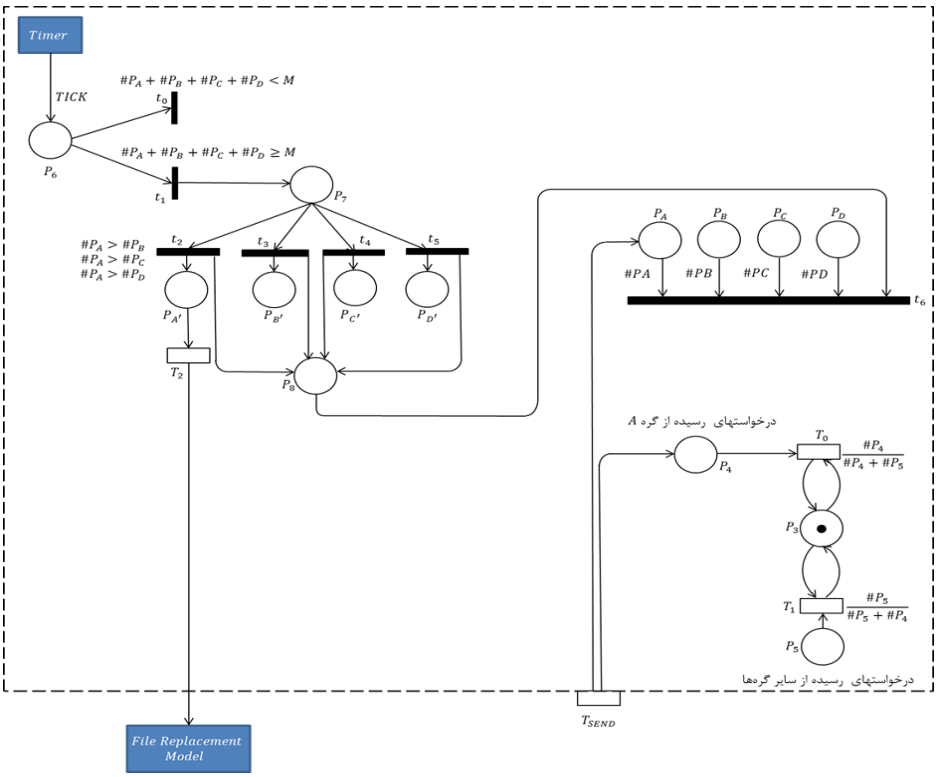


Figure 8. The SRN model of the root node in the best client policy

3.5.1 SRN model of the root node

The SRN model of the behavior of the root node in the best client policy is shown in Figure 8. The model consists of the following places and transitions which are described in the following. The place P_4 represents the number of requests received from the node A. Also, the place P_5 represents the number of requests received from the nodes B, C and D. The place P_3 represents the only server which is used to serve both the requests in places P_4 and P_5. The timed transitions T_0 and T_1 are for serving the requests respectively from the places P_4 and P_5. Once again, in order to approximate the FIFO scheduling, the probability of choosing the next request from the two places is determined such that the probability of choosing an item from a queue is proportional to the size of the queue. Therefore, the probability of choosing the request from the places P_4 and P_5 are respectively $\frac{\#P_4}{\#P_4+P_5}$ and $\frac{\#P_5}{\#P_4+P_5}$. The timed transitions T_0 and T_1 represent serving requests submitted respectively by this leaf and other leaves.

The places $P_{A'}$, $P_{B'}$, $P_{C'}$ and $P_{D'}$ represent the number of request received respectively from the nodes A, B, C and D. Once the replication timer ticks, a token is added to the place P_6 and depending on the current values of the places one of the two immediate transitions t_0 and t_1 becomes enabled. If $\#P_{A'} + \#P_{B'} + \#P_{C'} + \#P_{D'} < M$ the transitions t_0 and otherwise t_1 becomes enabled. Enabling of t_1 means that the number of requests for the files has reached beyond the threshold and a replication of the file should be created. When t_1 fires a token is added to the place P_7 and one of the immediate transitions t_2, t_3, t_4 and t_5 becomes enabled. The immediate transition t_2, t_3, t_4 and t_5 become enabled if respectively A, B, C and D are the best client. By firing each of these transitions a copy of the data item is created at the corresponding node. Also, a token is added to the place P_8. Once a token is added to the place P_8, the immediate transition t_6 becomes enabled. By firing this transition the variable related to the number of requests received from the nodes set to zero.

Conclusion

In this paper, we have proposed a framework for modelling and evaluation of data replication policies in data grid. In the proposed framework, the behavior of a typical data item and processing the access requests for the file are modelled. The proposed framework consists of separate sub-models for modelling workload generation, file replacement module, the replication timer and etc. We have used the framework to specify the best-client replication policy. Using the proposed framework, replication policies can be evaluated and compared with each other from the performance point of view.

Since the SRN model of the best-client data replication policy is built simply by putting the sub-models together, we can claim that the modelling a data replication policy using the proposed framework is quite simple.

References:

A.E. Baert, V.Boudet, A. Jean-Marie, "Performance analysis of data replication in grid delivery network", in: International Conference on Complex, Intelligent and Software Intensive Systems, 2008, pp.369-374.

Erling V. Matthiesen et al , Dependability Evaluation of a Replication Service for Mobile Application in Dynamic Ad-Hoc Networks, hal-00851873, version 1-23 Aug 2013.

F. I, K. C, S. C, T. S. CHERVENAK A, "The Data Grid: Towards an architecture for the distributed management and analysis of large scientific datasets", J.Net. Comput.Appl.23, pp.187-200, 2000..

I. Foster, C. Kesselman, The Grid: Blueprint for a new Computing Infrastructure, 2nd ed., The Morgan Kaufmann Series in Computer Architecture and Design, Morgan Kaufmann, 2003

I. Foster, K. Ranganathan, Decoupling computation and data scheduling in distributed data-intensive applications, in: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing, HPDC-11, IEEE, CS Press, Edinburgh, UK, (2002), pp. 352–358

K. Ranganathan, I. Foster, "Identifying dynamic replication strategies for high performance data grids", in: Proceedings of the Second International Workshop on Grid Computing, 2001, pp. 75-86.

Kishor S. Trivedi, Hairong Sun, Stochastic Petri Nets and Their Applications to Performance Aalysis of Computer Networks, National Science FoundTION UNDER Grant No. EEC9418765.

M. Humphrey, M. Thompson, K.R,Jackson, "Security for Grids", in: Proc. IEEE, special issue on Grid Computing, vol. 93, no. 3,2005, 644-652

M. Tang, B.-S. Lee, C.-K. Yeo, X. Tang, Dynamic replication algorithms for the multi-tier data grid, Future Generation Computer Systems 21(2005) 775–790

M.Cannataro, D.Talia, "The knowledge grid", Communications of the ACM, vol 46, no. 1, 2003, pp. 89-93.

M.Tang, B.-S. Lee, X.Tang, C.-K. Yeo,The impact of data replication of job scheduling performance in the data grid, Future Generation Computer Systems 22 (2006) 254–268..

Mohammad Shorfuzzaman, Peter Graham and Rasit Eskicioglu, Adaptive Replica Placement in Hierarchical Data Grids, 2010

Mohammad Shorfuzzaman, Peter Graham and Rasit Eskicioglu, Adaptive Replica Placement in Hierarchical Data Grids, 2010

Q. Gu, B. Chen, Y. Zhang, "Dynamic replica placement and location strategies for data grid", in: International Conference on Computer Science and Software Engineering, vol. 3, 2008, pp.35-40.

R. Kingsy Grace a, R. Manimegalai b, "Dynamic replica placement and selection strategies in data grids—A comprehensive survey ", 2014

R.Chang, P.H.Chen, "Complete and fragmented replica selection and retrieval in Data Grids", Future Generation Computer Systems, vol.23,2007, 356-546. .

R.S.Chang, J.S. Chang, S.Y.Lin,"Job sheduling and data replication on data grids", Future Generation Computer Systems vol. 23, 2007, pp. 846-860.

Rongfei Zeng, Yixin Jiang "Dependability Analysis of Control Center Networks in Smart Grid Using Stochastic Petri Nets"IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 9, SEPTEMBER 2012.

Ruay-Shiung C, Jih-Sheng C, Shing-Yi L, "Job scheduling and data replication on data grids", 2007.

Ruay-Shiung Chang, Jih-Sheng Chang, "Adaptable Replica Consistency Service for Data Grids", Third International Conference on Information Technology, (2006), pp.7-13.

S.-M. Park, J.-H. Kim, Y.-B. Go, W.-S. Yoon, Dynamic grid replication strategy based on internet hierarchy, in: International Workshop on Grid and Cooperative Computing, in: Lecture Note in Computer Science, vol. 1001, 2003, pp. 1324–1331

T.Bram.(2009). wiseGEEK [Online]. http://www.wisegeek.com/what-is-grid-computing.htm

Tehmina Amjad, M uhammad Sher, Ali Daud, A survey of dynamic replication strategies for improving data availability in data grids, in: Future Generation Computer Systems, elsevier, 2012, pp.337-349.

Yuzhong Sun and Zhiwei Xu,"Grid Replication Coherence Protocol", The 18th International Parallel and Distributed Processing Symposium, Santa Fe, USA, April 2004, pp.232-239