

A Survey for Replica Placement Techniques in Data Grid Environment

Alireza Souri

Department of Computer Engineering, College of Engineering, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran
Email: alireza.souri@iauasrb.ac.ir

Amir masoud Rahmani

Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
Email: rahmani@srbiau.ac

Abstract—In data grids, data replication on variant nodes can change some problems such as response time and availability. Also, in data replication, there are some challenges to finding the best replica efficiently in relation to performance and location of physical storage systems. In this paper, various replica placement strategies are discussed. These replica placement strategies are available in the works. Replica placement contains recognizing the best possible node to duplicate data based on network latency and user request. These strategies measure and analyze different parameters such as access cost, bandwidth consumption, scalability, execution time and storage consumption. This paper also analyses the performance of various strategies with respect to the parameters mentioned above in data grid.

Index Terms—Dynamic Replication, Replica Placement, Data Grid.

I. INTRODUCTION

In a developed category of data exhaustive methodical and commercial presentations such as high energy element physics and astronomy [1], large volumes of data sets may be created, accessed and shared, from different places with various quality of service requirements. Data Grid [2] make available organically distributed resources for large-scale data exhaustive applications that create large data sets. However, organization of widely distributed massive data gives increase to many design issues such as data consistency, security, fast and reliable access and access authorizations [3].

One of the important experiments in data replication techniques is replica placement. Replica placement is the problem of placing duplicate copies of data in the most suitable node in the data grid. The replica placement can be logically divided into four strategy in data grid: multi-tier [4], graph topology [5, 6], Hierarchical [7-11] and peer to peer [12].

There are two types of data replication techniques, namely, static and dynamic. In static data replication, the number of replicas to be created and the nodes where

replicas should be placed are decided statically during grid setup time. The static replication strategies are simple to implement but not frequently used because it does not support data duplication during job execution. Static replication strategies have the advantages of reduced overhead and quick job scheduling [13, 14].

Dynamic replication [15] can adapt changes based on user requests, storage capacity and bandwidth. Dynamic replication schemes are capable of making intelligent decisions to place data in the grid based on storage capacity and node availability. Replicas are created in a centralized or distributed fashion. In a centralized scheme, replicas are created only in the head node, whereas, in a distributed scheme, replicas are created in a few selective nodes in addition to the head node. We analyze summary of dynamic replica placement techniques in some recent research works.

The reset of this paper is: section II illustrates some strategies of the replica placement in distributed technique. Section III shows the conclusion of these summaries.

II. REPLICA PLACEMENT TECHNIQUES

There are two techniques for replica placement as follow: centralized and distributed that are further classified according to the type of network used for implementation. In this paper we analyze distributed dynamic replication placement technique in data grids.

Three different dynamic replica placement algorithms are considered which are combined with the four scheduling strategies.

Permanent replicas: Process always having a replica. Permanent replicas can be considered as the initial set of replicas that constitute a distributed data store. In many cases, the number of permanent replicas is small.

Server-initiated replica: Process that can dynamically host a replica on request of another server in the data store. In contrast to permanent replicas, server-initiated replicas are copies of a data store that exist to enhance performance and which are created at the initiative of the (owner of the) data store.

Client-initiated replica: Process that can dynamically host a replica on request of a client (client cache). An important kind of replica is the one initiated by a client. Client-initiated replicas are more commonly known as (client) caches. In essence, a cache is a local storage facility that is used by a client to temporarily store a copy of the data it has just requested.

They show that when there is no replication, simple local scheduling performs best. However, when a replication is used scheduling jobs to sites containing the required data is a better approach.

In this survey we show some important replica placement strategies and compare performance of them.

A. Graph topology strategy

M. Tu et al. [6] have studied the problem of optimal allocation of sensitive data objects. The proposed solution adopts a two layered grid topology. The upper layer is a multiple cluster network which can be represented by a general graph. A Directed Acyclic Graph (DAG) is used to denote the topology within each cluster (see in Figure 1). The authors are disintegrated replica allocation problem into two sub problems: first, the Optimal Inter cluster Resident Set Problem (OIRSP), which determines the cluster that needs shared replicas and second the Optimal Intra-cluster Share Allocation Problem (OISAP), which determines the number of shared replicas needed in a cluster and their placements. To find the Optimal Inter cluster Resident Set Problem the Minimal Spanning Tree (MST) is used. They are used SDP – Tree algorithm to find the number of replicas to be placed in a node with minimal communication cost.

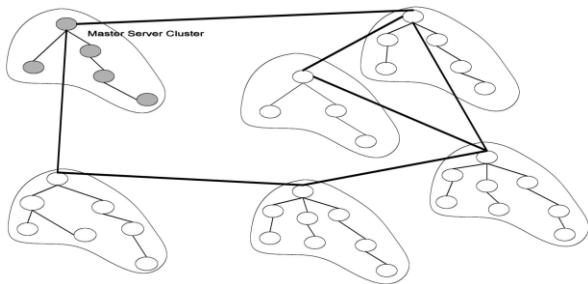


Fig. 1. A graph system topology [6]

B. Multi-tier architecture strategy

Ranganathan et al. [4] have discussed three variant kinds of access designs with respect to file locations namely, temporal locality, geographical locality and spatial locality. Temporal locality determines the request pattern in such a way that the recently accessed file is likely to gain access again. Geographical locality is based on the locality of the client. It is based on the notion that a file requested by a client is more likely to be requested by the neighboring clients. The spatial locality exploits the file locality i.e., a file in a particular locality is likely to be accessed again. Random access pattern is the one in which files are accessed with no definite pattern. The simulation starts with topology specification and is

followed by a request for the closest replica site. The nearest best replica is located by using a heuristic method. The file transfer is initiated based on the client request and the response time is recorded. The best possible replica is the one with the lowest response time. Six different strategies, such as, caching, best client, cascading replication, plain caching, caching plus cascading replication and fast spread were adopted for replica placement in [4]. A comparative study to evaluate the performance of the above mentioned strategies is also done based on the bandwidth consumption and the average response time. Caching is the client side phenomenon in which the file requested by the client is stored locally for future use. In the best client strategy, the history of each file is maintained with details such as, number of requests and clients who requested the file. In cascading strategy, the entire grid is divided into several levels. Initially the replica is created only at the root i.e. at the first level. As the root gets filled up, replicas are created in the next, cascading levels based on a threshold. In the case of plain caching, the client stores the requested file locally. When cascading is combined with caching strategy, a locally cached file which is frequently requested by clients is cascaded down in the hierarchy by the server. In the fast spread strategy, a replica is created in each tier in the path to the client. It has the advantage of less bandwidth requirement when compared to the cascading strategy. But it has the disadvantage of high storage space requirement. Fast spread works fine for random access pattern, whereas, the cascading is suitable if the average response time is the prime consideration. The work done in [4] helps in understanding how the replica creation and data movement impact the performance of the grid.

C. Tree architecture strategy

Perez et al. [16] have proposed a Branch Replication Scheme (BRS) which optimizes the storage requirement. It is achieved by creating sub replicas and using parallel I/O techniques. In BRS, sub replicas are created in a hierarchical tree topology, not overlapping with each other. BRS reduces data access time and maintains consistency (shown in Figure 2). In this model, all replicas of a single file are updated in parallel with the updating done in the master. The simulated results show that the proposed BRS scheme increases the performance of the grid in terms of the data access time.

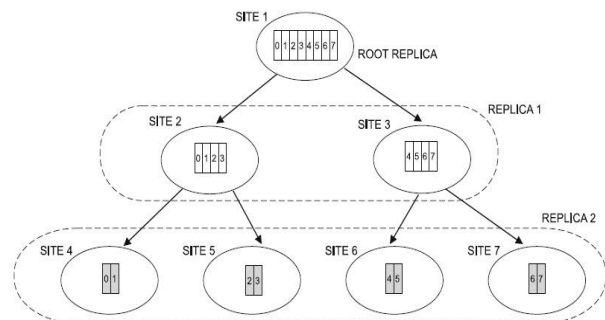


Fig. 2. A tree branch replication strategy [16]

D. Free-scale topology strategy

Chen et al. [5] have proposed a replica creation algorithm which reduces the makespan and data storage consumption with efficient load balancing capability. The proposed replica creation algorithm selects nodes with optimized load, high degree and greater frequency for replication (see in Figure 3).

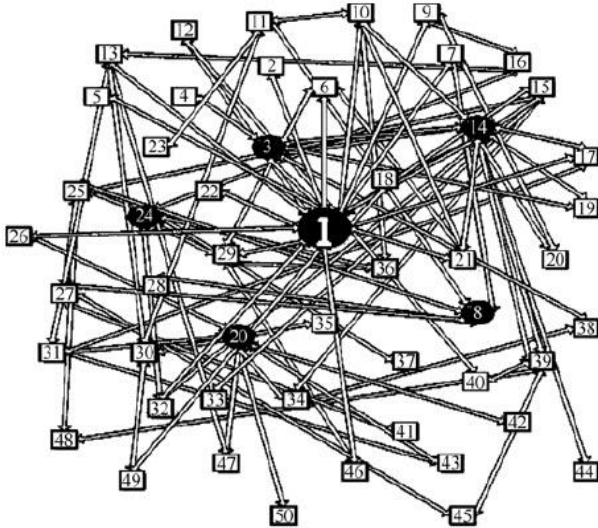


Fig. 3. A free-Scale data grid topology

The authors presented two pools in order to facilitate the node selection: first the degree-based-candidate-pool with optimized network load and second the frequency-based-candidate-pool with optimized network load. The degree-based-candidate-pool comprises nodes with high degree and the frequency-based-candidate-pool comprises nodes with high frequency. A node is selected for replica creation if it is available in both the pools mentioned above. If not, the cost for each node is computed as the sum of storage cost and file transfer cost. If there is no suitable node with high degree and frequency then the minimum cost node from these pools is selected for data replication.

E. Hierarchical architecture strategy

The BHR algorithm in [11], groups all grid sites in the same network region. More replicas are created in the sites that have broad bandwidth in order to serve replica requests. In the modified BHR algorithm, frequently accessed files are replicated and stored in local region for future requirement. The access history of every site, which contains details of frequently accessed files, is maintained in the region header (shown in Figure 4). Simulation results show that the modified BHR algorithm provides increased data availability than the original BHR algorithm presented in [11].

Mansouri et al. [10] have proposed a distributed replication algorithm named Dynamic Hierarchical Replication Algorithm (DHR). The DHR algorithm extends the proposed algorithm in [17]. DHR selects

replica location based on the data transfer time and request–waiting-time in storage during job execution. The DHR algorithm consists of the following three steps: replica selection, replica decision and replica replacement. During replica selection, it selects the site which has the least number of requests. If the requested replica is not available in the local site, then, replication takes place. If the local site is not having sufficient storage space and if the requested replica is not available in the same LAN, then replica replacement is carried out.

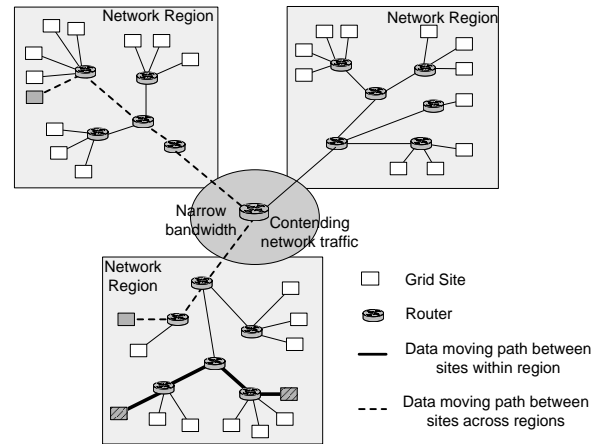


Fig. 4. Replication Strategy based on Bandwidth Hierarchy[11]

The OporSim simulation results show that DHR algorithm prevents unnecessary replication and takes less execution time when compared to other strategies such as LRU, LFU, BHR [11] and 3-Level Hierarchical Algorithm (3-LAH) [17]. It is observed that the DHR algorithm outperforms the above mentioned strategies by 37% and is also suitable for grid sites with small storage size [10].

The combination of Modified DHR Algorithm (MDHRA) and Combined Scheduling Strategy (CSS) is proposed by mansouri et al. in [18]. MDHRA considers three factors for replacing replicas: (1) the last time the replica was requested; (2) number of accesses and (3) size of replica. It computes the response time based on the data transfer time, storage access latency, the unprocessed replica requests and the distance between nodes. The computed response time is used to select the best replica location. The CSS scheduler uses a hierarchical grid structure and are two steps decision process: (1) best region selection and (2) assignment of job to the site with minimum combined cost within the best region. The proposed MDHRA strategy is compared with LFU, LRU, BHR, Modified BHR, 3LHA and DHR and was proved the best. It minimizes the bandwidth consumption thus reducing the network traffic.

Effective scheduling reduces the traffic across the internet through appropriate job dispatching strategies. Chang et al. have proposed efficient Hierarchical Cluster Scheduling (HCS) and Hierarchical Replication Strategy (HRS) that provide high bandwidth utilization, faster data access and low intercommunication cost in [7]. HCS schedules a job based on data location, access cost and the number of jobs in the queue at a particular node. A

detailed performance analysis based on the job execution time, communication overhead and resource usage is also done in [7]. The replication strategy in HRS uses the region optimizer to select the best cluster and the best site with less inter cluster communication time and less intra-cluster communication cost respectively. The queue access cost of the proposed method is also compared with that of Least Recently Used (LRU), Least Frequently Used (LFU), and Bandwidth Hierarchical Based Replication (BHR). It is absorbed that the combined HCS and HRS replication strategy reduces the file transmission time and thereby improves the throughput by reducing the total job execution time.

F. Peer-to-peer architecture strategy

Ranganathan et al. [19] have proposed a dynamic model-driven replication strategy which increases the data availability in large scale peer to peer networks. Each peer creates replicas automatically in distributed manner. The proposed replica placement framework maintains a threshold level of availability at all times. The parameters considered in this work are: (1) the probability of a node that may be alive; (2) the accuracy of the replica location service; (3) replica transfer cost between node A and node B; and (4) storage cost for file F at node N. The best candidate node to store a replica is the one which maximizes the difference between the transfer cost and benefit. As replica decision is decentralized, more than one peer can create same

replicas to get the desired availability threshold. It wastes the storage space. The limitations of this strategy are: (1) unlimited storage space is assumed to be available, but it is not the case in real scenarios; (2) the overhead to invoke replica location service for each replication.

The result of various replica placement strategies are presented in Table 1.

III. CONCLUSION

In this paper we present a survey on replica placement for dynamic data grid. Different replica placement strategies are proposed by researchers. In dynamic grid arrangement, the user can join and leave the network at any point of time. So, there is no specific grid topology used for the dynamic data grid. The graph or hierarchy network topology is well suited for dynamic data grid. The proposed methods available in the literature addressed one or more of the following issues: fault tolerance, scalability, reliability and load balancing. Also Bandwidth maintenance is extra important issue which is optimized by a few replication techniques. There are no sufficient researches for replication techniques that focus on consistency of data replication. Most of the techniques proposed [7, 10, 18, 19] are based on simulation, not on formal verification and modeling implementation. Implementing these replication techniques in real time, verifying and testing the real efficiency is an interesting open problem in future work.

Table 1. Summary of various data replication placement strategies

Methodology	Architecture	Replication classification	Evaluation parameters	Simulator used	Performance
Model Driven Replication [19]	Peer to Peer	Distributed	Availability Threshold, Accuracy	Not Mentioned	It is able to predict the required number of replicas in the system with moderate accuracy.
BHR Region Based Algorithm [11]	Hierarchical	Distributed	Storage Element Utilization, Total number of Replication, Average Job Time, Network Utilization	OptorSim	BHR takes 30% lesser execution time than LRU.
No Replication, Best Client, Cascading Replication, Plain Caching, Caching plus Cascading Replication and Fast Spread Strategy [4]	Multi-tier	Distributed	Bandwidth Consumption, Response Time	PARSEC Tool	For getting faster response time Cascading is best. For conserving bandwidth Fast Spread is best.
A Hierarchical Scheduling and Replication Strategy [7]	Hierarchical	Distributed	Mean Job Time, Total Job Time, Size of SE	OptorSim	13% better than LRU and BHR
OIRSP and OISAP using Shamir's algorithm [6]	Graph	Distributed	Communication Cost	OptorSim	Reduced Communication cost than K replication, No replication and Complete replication
Branch Replication Scheme (BRS) [16]	Tree	Distributed	Bandwidth Consumption, Response Time, Network Latency	Omnet++	Improved data access Performance than hierarchical (HRS) and server-directed (SDRS)

Dynamic Multi-Replica Creation (DMRC) [5]	Free-scale	Distributed	Makespan, Data Storage Consumption	OptorSim	DMRC reduces makespan and Storage consumption than Always Replicate, EcoMode, EcoMode-zipf and Simple.
Dynamic Hierarchical Replication (DHR) algorithm [10]	Hierarchical	Distributed	Mean Job Time, Size of SE, Size of File	OptorSim	DHR outperforms the methods such as LRU, LFU, BHR and 3LHA.
Scheduling Strategy (CSS) and Modified Dynamic Hierarchical Replication Algorithm (MDHRA) [18]	Hierarchical	Distributed	Mean Job Time, Number of intercommunications, Network Utilization, Bandwidth of Inter-Region link	OptorSim	It outperforms LRU, LFU, BHR, MBHR, 3LHA and DHR.

REFERENCES

- [1] V. Andronikou, K. Mamouras, K. Tserpes, D. Kyriazis, and T. Varvarigou, "Dynamic QoS-aware data replication in grid environments based on data "importance", " *Future Generation Computer Systems*, vol. 28, pp. 544-553, 3// 2012.
- [2] A. Souri and N. Jafari Navimipour, "Behavioral modeling and formal verification of a resource discovery approach in Grid computing," *Expert Systems with Applications*, vol. 41, pp. 3831-3849, 6/15/ 2014.
- [3] S. Vazhkudai, S. Tuecke, and I. Foster, "Replica selection in the Globus Data Grid," in *Cluster Computing and the Grid*, 2001. Proceedings. First IEEE/ACM International Symposium on, 2001, pp. 106-113.
- [4] K. Ranganathan and I. Foster, "Identifying Dynamic Replication Strategies for a High-Performance Data Grid," in *Grid Computing — GRID 2001*. vol. 2242, C. Lee, Ed., ed: Springer Berlin Heidelberg, 2001, pp. 75-86.
- [5] D.-w. Chen, S.-t. Zhou, X.-y. Ren, and Q. Kong, "Method for replica creation in data grids based on complex networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 17, pp. 110-115, 8// 2010.
- [6] T. Manghui, L. Peng, I. L. Yen, B. Thuraisingham, and L. Khan, "Secure Data Objects Replication in Data Grid," *Dependable and Secure Computing*, IEEE Transactions on, vol. 7, pp. 50-64, 2010.
- [7] R.-S. Chang, J.-S. Chang, and S.-Y. Lin, "Job scheduling and data replication on data grids," *Future Generation Computer Systems*, vol. 23, pp. 846-860, 8// 2007.
- [8] Z. Hai, Z. Zehua, and Z. Xuejie, "A Dynamic Replica Management Strategy Based on Data Grid," in *Grid and Cooperative Computing (GCC)*, 2010 9th International Conference on, 2010, pp. 18-23.
- [9] K. Sashi and A. S. Thanamani, "Dynamic replication in a data grid using a Modified BHR Region Based Algorithm," *Future Generation Computer Systems*, vol. 27, pp. 202-210, 2// 2011.
- [10] N. Mansouri and G. H. Dastghaibfard, "A dynamic replica management strategy in data grid," *Journal of Network and Computer Applications*, vol. 35, pp. 1297-1303, 7// 2012.
- [11] S.-M. Park, J.-H. Kim, Y.-B. Ko, and W.-S. Yoon, "Dynamic Data Grid Replication Strategy Based on Internet Hierarchy," in *Grid and Cooperative Computing*. vol. 3033, M. Li, X.-H. Sun, Q. Deng, and J. Ni, Eds., ed: Springer Berlin Heidelberg, 2004, pp. 838-846.
- [12] Z. Challal and T. Bouabana-Tebibel, "A priori replica placement strategy in data grid," in *Machine and Web Intelligence (ICMWI)*, 2010 International Conference on, 2010, pp. 402-406.
- [13] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, et al., "Giggle: A Framework for Constructing Scalable Replica Location Services," in *Supercomputing*, ACM/IEEE 2002 Conference, 2002, pp. 58-58.
- [14] O. Tatebe, Y. Morita, S. Matsuoka, N. Soda, and S. Sekiguchi, "Grid Datafarm Architecture for Petascale Data Intensive Computing," in *Cluster Computing and the Grid*, 2002. 2nd IEEE/ACM International Symposium on, 2002, pp. 102-102.
- [15] N. Saadat and A. M. Rahmani, "PDDRA: A new prefetching based dynamic data replication algorithm in data grids," *Future Generation Computer Systems*, vol. 28, pp. 666-681, 4// 2012.
- [16] Jos, #233, M. P, #233, rez, #233, et al., "Branch replication scheme: A new model for data replication in large scale data grids," *Future Gener. Comput. Syst.*, vol. 26, pp. 12-20, 2010.
- [17] A. Horri, R. Sepahvand, and G. Dastghaibfard, "A hierarchical scheduling and replication strategy," *Int J Comput Sci Netw Secur*, vol. 8, pp. 30-35, 2008.
- [18] N. Mansouri, G. H. Dastghaibfard, and E. Mansouri, "Combination of data replication and scheduling algorithm for improving data availability in Data Grids," *Journal of Network and Computer Applications*, vol. 36, pp. 711-722, 3// 2013.
- [19] K. Ranganathan, A. Iamnitchi, and I. Foster, "Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities," in *Cluster Computing and the Grid*, 2002. 2nd IEEE/ACM International Symposium on, 2002, pp. 376-376.



Alireza Souri received his B.Sc. in Computer Engineering from University College of Nabi Akram, Iran, 2011. Currently, he receives M.Sc. in Software Engineering from East Azarbaijan Science and Research Branch, Islamic Azad University in Iran. His main interests are formal verification, behavioral modeling and analyzing Distributed Databases, Multilayer systems, Antivirus systems, Grid computing, and Real-time systems. He is member of The Society of Digital Information and Wireless Communications.



Amir Masoud Rahmani received his B.S. in computer engineering from Amir Kabir University, Tehran, in 1996, the M.S. in computer engineering from Sharif University of technology, Tehran, in 1998 and the Ph.D degree in Computer Engineering from IAU University, Tehran, in 2005. He is an associate professor in the department of

computer and mechatronics engineering at the IAU University. He is the author/co-author of more than 140 publications in technical journals and conferences. He served on the program committees of several national and international conferences. His research interests are in the areas of distributed systems, ad hoc and sensor wireless networks, and evolutionary computing.