A Survey of Dynamic Replication Strategies in Distributed Systems

Naveen Dogra
Panjab University, SSG
Regional Centre, Hoshiarpur

Sarbjeet Singh
UIET, Panjab University Chandigarh

ABSTRACT
In today's world, effective management and quick retrieval of data is an important issue. It is very challenging task to provide maximum data availability and fault tolerant data access in the distributed systems as it scales over different geographical areas. To handle this issue data has to be replicated on large number of different sites. Data replication is a practical and effective method to enhance the performance and reliability in distributed systems by making multiple copies of same data. In this paper a study is conducted to examine already proposed data replication techniques in distributed systems and we found that replication techniques are designed based on various parameters such as availability, scalability, bandwidth consumption, access time and fault tolerance etc. A comparison has been done between different replication strategies based on different parameters. The paper also discusses future work in the area of replication in the distributed systems.

Keywords
Distributed Computing, Data Replication, Data Availability, Reliability, Bandwidth Consumption.

1. INTRODUCTION
These days data intensive applications such as scientific research and other commercial applications produce huge amount of data and this data is used by the user for making important decisions [1]. Handling this data in centralized way is not an effective method as it leads to large access latency. Moreover the centralized methods make central server heavily loaded [2]. So, distributed approach is preferred for data handling in distributed systems. In distributed systems, it is a challenging task to provide high availability, high fault tolerance and efficient access to required data because of its dynamic nature [1]. Data Replication is very useful strategy in distributed systems for improving data availability and reliability, reducing waiting time for user, improving fault tolerance and minimizing bandwidth consumption by providing multiple replicas of data on different nodes [3-8]. If any data site fails due to natural disasters or network failures, a system can still operate using copy of replica, thus increasing data availability. Another reason for replication of data or file is to increase the performance when the distributed system needs to scale in size in terms of number of nodes or geographical area. Scaling in number is required when the number of processes accessing a data item increases and there is only a single server managing this task. In this case the performance can be increased by replicating the server and dividing the work between them. Scaling in size in terms of geographical area may also require replication of data. Placing a copy of data near to the process using that data decreases access time and reduces bandwidth consumption [4]. With the development of various technologies, data replication can be studied in various ways in distributed systems. We can divide data replication algorithms in two categories: static replication and dynamic replication. In a static replication algorithms the replica numbers and their location is predetermined. On the other hand, in dynamic replication algorithms the decision of creation and deletion of replica depends upon the work load and changing environment conditions. Because of intelligent decision making and placing the replica according to environment conditions makes dynamic replication strategies better than the static replication strategies. Further, the dynamic replication strategies do not create the replica of all the files, only popular files are replicated. So, to get the high availability, fault tolerance and efficiency, the important data files should be dynamically adjusted. Although data replication has many advantages but there are some drawbacks as well. Too much replicas may not always increases the data availability but sometimes brings unnecessary spending and it is again a challenge to place the new replicas on different nodes according to the current environments conditions in the distributed systems such as cloud systems. Keeping above drawbacks in mind to achieve better dynamic replication systems, three important problems must be solved. 1) It is important to find the data which should be replicated in distributed system and when, so as to meet the user requirements such as increase in data access speed and reduction in waiting time. 2) To achieve system availability requirement, it is important to find the number of new replicas that should be created in the system 3) To reduce the bandwidth requirements and get the maximum job execution rate, it is important to determine where we should place the replica? All these are important concerns that need to be addressed [9].

2. LITERATURE SURVEY
In large scale data intensive applications, data needs to be replicated to get better availability and fault tolerance. Replicating the data at more than one locations or sites also enhances the reliability of the data. The multiple copies of the data may also help to protect the data from corruption. In the grid to get efficient and fault tolerant data access, the data need to be replicated [10]. Data replication techniques also help in load balancing by making the multiple copies of data and placing the copies to multiple sites. The main aim of replication is to increase the data availability. The data replication schemes studies in the data grid are available in [11], [12]. The replication strategies also increase the performance of the system, the effect of replication in the performance of system and scalability aspects is studied in [13-15]. One of the major problems in making the replica is to keep consistency. It means when one copy of replica is updated then all others copies need to be updated otherwise all the replicas may not be same and leads to ambiguous data or files.

Several researchers, both from academia and the industry are working in the area of replication and consistency management in distributed systems. This section describes recent developments in this area with the objective to identify and categorize them for easy reference.

K. Ranganathana and Foster (2001) developed replication strategies using the data access patterns [11]. They proposed six different replication strategies and the architecture used by them is multi-tier architecture. Main focus of these strategies is to reduce the access latency and bandwidth consumption. The proposed replication strategies are "No Replica", "Best Client", "Cascading Replication", "Plain Caching", "Caching plus Cascading Replica" and "Fast Spread". In their study they introduced three types of access patterns. First type is "temporal locality" it suggests that most recently accessed file has the maximum possibility of requesting again. The second type of access pattern is "geographical locality". It suggests that file recently accessed may be accessed by the adjacent client too in the same region. Third and the last access pattern considered is "Spatial locality". It suggests that recently accessed file may be accessed in the near future. They evaluated all six strategies using different access patterns and found that Cascading and Fast Spread strategies performed best in the simulations. Cascading replication performed well for reducing the access latency and Fast Spread replication performed well when main aim is to reduce bandwidth consumption, but it wastes lot of storage also [16].

K. Ranganathan et al. (2002) [17] proposed a replication strategy named Dynamic Model-Driven Replication strategy. The architecture used by them is peer to peer architecture and the replication decision is taken in decentralized fashion in this strategy. The main focus of this strategy is to provide the data availability in peer to peer community. To improve the availability, the peer can produce the replica automatically in decentralized way. Since the peer can independently take the decision of replication of file, this may leads two peers to create the replica of the same file. Every peer has its set of tools to check the status of file and can take replication decision to maintain the availability level of file at certain threshold level. The advantage of this technique is that there is no single point of failure since the decision is taken by the independent peer regarding replication. But it also leads to wastage of storage space since more than required number of replicas can be created in this technique. The other objectives of this approach are to determine the appropriate host for new replica and also to find the optimal number of replicas. The cost and benefit are the two parameters which judge the best client, the difference between cost and benefit should be large. The limitation of this strategy is that the strategy considers that there is unlimited storage for creating replica, but it is not possible in practical. Another overhead is that replica location service must be invoked every time while creating replica [17].

S. Park et al. (2004) [18] proposed a replication technique that uses network bandwidth as the basis of replication and named the technique as Bandwidth Hierarchy based Replication. This data replication strategy uses data access pattern on network level. The main focus of this technique is to reduce data access time by avoiding the congestion in the network. The site is divided into different regions. It is obvious that network bandwidth within the region is much more than between the regions. So file finding in the same region takes less time as compared to between the regions. As in the same region, the number of routers between the paths is less and network bandwidth is more whereas the numbers of routers between the regions are more and network bandwidth is less. To decrease the data access time and increase network level locality, the BHR technique creates replicas within the region. Region optimizer keeps the count of number of accesses of the file and then takes the decision regarding replicating the file on the basis of storage availability. When a request for a file is made, firstly it is checked in the requesting site and if it is not found then it is fetched from the other site and after processing the file, the decision regarding replication is taken by keeping local storage into consideration. If the available space is not enough then the room for new file is created by removing the existing files. For doing so, first it is checked that same file may not be stored on the other sites of the same region. If same file exists, then file is not stored. If the file has to be stored on to the specified site then the same file has to be deleted from the local storage of other sites to create the space for storage. Second step is to store all files in sorted order in the storage element on the basis of least frequently used. New replica can be stored after deleting all the files with access latency less than the file that has to be replicated. When the enough space is created then the replication operation is performed. The first step is used to avoid the duplication of data and the second step is used to make the replica of the popular files only [18].

Jose M. P. et al. (2010) [10] propose a replication scheme called Branch Replication Scheme (BRS). The architecture of BRS is hierarchical in nature. To manage data on the grid the data replication scheme plays an important role [19], [20]. The storage requirements of traditional data replication schemes are very large. BRS emphasizes on the optimal storage consumption. Unlike the traditional data replication schemes where the whole replica is placed on the each site, the BRS stores only the sub part of the replica on the site and also the parallel access to this sub replica is possible. This increases the data access performance [21]-[23]. The previous schemes of data replication do not allow the efficient data modifications as this leads to the data consistency problem. The Branch Replication Scheme maintains the consistency of replicas while updating. This scheme also provides fault tolerance, improved performance and scalability. Authors have also proposed the naming scheme and scheme of replica updating. Simulation results show that BRS scheme provides the better data access performance and scalability than the other data grid replication schemes such as hierarchical replication scheme and server directed data replication scheme. The Branch Replication Scheme performs better than Hierarchical Replication Scheme for both read and writes operation on all sizes of files [10].

Shorfuzzaman et al. (2008 and 2010) proposed a dynamic replication strategy for placing the replica on hierarchical data grid. This scheme is based on the popularity of file and named as Popularity Based Replica Placement (PBRP) strategy. The popularity of the file is calculated by maintaining the record of data accesses by the clients. It is assumed that the popular files may be used in near future. The algorithm is initiated at regular interval to check the dynamics of access patterns. The availability and access rate of data can be improved by planned placement of the replica. The PBRP algorithm became efficient by determining the threshold value of file popularity. The main focus of algorithm is to reduce job execution time, proper bandwidth consumption and proper storage utilization. It is found that the job execution of this algorithm is better than "best client" and "caching" algorithms. Authors have also proposed Adaptive-PBRP (APBRP) scheme. In this scheme the popularity threshold value can be calculated by data request arrival rate. The PBRP and APBRP perform better than other replication strategies with respect to bandwidth consumption and job execution time [24], [25].

Q. Wei et al. (2010) [8] present a data replication scheme for heterogeneous systems. The main focus of the scheme is to provide improve data availability by cost effective way and also improve the performance by keeping the balanced workload. The replication scheme cost-effective dynamic replication management scheme (CDRM) works on the basis of blocking probabilities of the nodes. The blocking probability of the node is calculated by using the arrival rate of request and time required to serve the request. This scheme maintains the relation between the availability and the number of replicas. If the availability of the specified replica is less than the user requirements then the file is replicated to maintain the availability level. In this way CDRM calculates the total number of replicas required to improve the performance of the system. Replica placement is also very important issue in this scheme. Replica placement is based on the capacity of the node in the heterogeneous environment and the blocking probability of the node. Effective replica placement helps in maintaining the workload in the heterogeneous system. This scheme is implemented with the help of Hadoop Distributed File System (HDFS). The name node keeps the information of the blocking probability of the node by using the data structure B+ tree. Each node will give the information about its blocking probability to the name node. Experimental results demonstrate that the CDRM scheme improves the data availability, improves access latency and also provides load balancing in cloud systems and also proves that the scheme performs well than replication management of HDFS [8].

M. Bsoul et al. in 2010 [26] proposed a new dynamic replication strategy called Enhanced Fast Spread replication technique for data grid environment based on the Fast Spread replication technique. The only difference between above two strategies is that the EFS consider the “importance” of replica before replacement of the new replica. The topology used in EFS scheme is graph topology with one server node and many client nodes. The EFS technique considers many parameters such as frequency of request, last time of replica request and size of replica for making the replication decision. All the data is stored on the server node as the client node has limited amount of storage. When there is a file request, it is first searched locally. If not found locally, then client searches the file from the other nearby clients. For this, client passes through the shortest path to find the requested file. EFR technique replicates the files on all nodes along the path and if there is shortage of space then it makes room for replica using Least Recently Used (LRU) or Least Frequently used (LFU) techniques. The EFR scheme creates replica only when the new replica has greater value than the group of replica value. To calculate the value of new replica and group replica the parameters like size of replica, number of requests must be considered. Experimental results show that EFS scheme reduces response time and provides better bandwidth consumption than FS technique [26].

Sun D.W. et al. (2012) [9] proposed a replication scheme for multitier hierarchical architecture in cloud system. The proposed dynamic data replication scheme is used to improve the data availability on the basis of temporal locality. The temporal locality suggests that the popularity of the data file that is recently accessed is more than other file and the chances of the file used in near future is more. If the popularity of the file passes the dynamic threshold value then the replication operation is triggered for the file. The availability of the file is increased by placing the multiple replicas on the multiple sites, so that the required process gets the replica from the nearby or adjacent site. To make the optimal use of storage, the replica of the data file that has the least popularity, should be deleted. The dynamic replication strategy solved the issues such as which file and when to replicate that file to improve data access time. It finds the appropriate number of replicas to maintain the availability, and where to place the replica to balance the load. The proposed scheme is evaluated in the CloudSim environment and experimental results showed that the scheme provides improved availability, reduced waiting time and load balancing in cloud systems [9].

3. COMPARISION

The comparison among different replication techniques in distributed systems is shown in the Table 1.

Table 1. Comparison of Replication Techniques

Feature	**[17]**	**[18]**	**[10]**	**[24]**	**[8]**	**[26]**	**[9]**
Year	2002	2004	2010	2008	2010	2010	2012
Replication Decision	Decentralized	Decentralized	Centralized	Centralized	Decentralized	Centralized	Decentralized
Architecture	P2P	General	Multi-tier	Multi-tier	HDFS	Graph	Multi-tier Hierarchal
Availability Improvement	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Response Time Reduction	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Scalability	No	No	Yes	No	Yes	No	No
Reliability	No	No	No	No	Yes	No	Yes
Load balancing	No	No	No	No	Yes	No	Yes
Storage capacity	Unlimited storage	Limited storage	Limited storage	Limited storage	Limited storage	Limited storage	Limited storage
Access Cost Reduction	No	Yes	No	Yes	Yes	Yes	No
Considers Bandwidth Consumption	No	Yes	No	No	Yes	Yes	Yes

4. SUMMARY & FUTURE SCOPE

The paper discusses the survey of various replication techniques in the distributed systems. Every strategy has presented its own terms for evaluation. Each technique addresses some of the issues and tries to resolve that. We have done comparison among different replication techniques and found that much work is required in data replication in distributed systems, particularly in cloud systems.

5. REFERENCES

[1] Ming Tang, Bu-Sung Lee, Xueyan Tang and Chai-Kiat Yeo, "The impact of data replication on job scheduling performance in the Data Grid" , Future Generation Computer Systems, vol.22, pp.254–268, 2006.

[2] N. Mansouri and G.H. Dastghaibyfard, "A dynamic replica management strategy in data grid", Journal of Network and Computer Applications, vol. 35, pp. 1297–1303, 2012.

[3] M. K. Hussein and M.H. Mousa, " A Light-weight Data Replication for Cloud Data Centers Environment", in International Journal of Engineering and Innovative Technology, Vol. 1, no. 6, pp. 169-175, 2012.

[4] A.S. Tanenbaum and M.V. Steen. Distributed systems principle and paradigms. PHI, 2010.

[5] Amjad, T., Sher, M., & Daud, A. (2012). A survey of dynamic replication strategies for improving data availability in data grids. Future Generation Computer Systems, 28(2), 337-349.

[6] K. Shvachko et al., "The Hadoop Distributed File System", in Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies, IEEE Computer Society, pp. 1-10, 2010.

[7] A. Dogan "A study on performance of dynamic file replication algorithms for real-time file access in Data Grids", Future Generation Computer Systems, vol. 25, no. 8, pp. 829-839, 2009.

[8] Q. Wei et al., "CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster", in IEEE International on Cluster Computing, pp. 188 - 196, 2010.

[9] D. W. Sun, G. R. Chang et al., "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments", Journal of Computer Science And Technology, vol.27,no. 2,pp.256-272, 2012.

[10] J. M. Perez et al., "Branch replication scheme: A new model for data replication in large scale data grids", Future Generation Computer Systems, vol. 26, pp. 12-20, 2010.

[11] K. Ranganathan, I. Foster, "Simulation studies of computation and data scheduling algorithms for Data Grids", Journal of Grid Computing, pp.53-62, 2003.

[12] L. Guy, P. Kunszt, E. Laure, H. Stockinger, K. Stockinger, "Replica Management in Data Grids" Technical Report, GF5 Working Draft, 2002.

[13] H. Stockinger, A. Samar, B. Allcock, I. Foster, K. Holtman, B. Tierney, "File and object replication in data grids", in 10th IEEE Symposium on High Performance and Distributed Computing, pp. 305–314, 2001.

[14] B. Allcock, et al. "Secure, efficient data transport and replica management for high performance data intensive computing", IEEE Mass Storage Conference, 2001.

[15] A.L. Chervenak, N. Palavalli, S. Bharathi, C. Kesselman, R. Schwartzkopf, "Performance and scalability of a replica location service", in High Performance Distributed Computing Conference, HPDC-13, Honolulu, HI, pp. 182-191, 2004.

[16] K. Ranganathan, I. Foster, " Identifying dynamic replication strategies for a high performance data grid", in Proceeding of the second International Grid Computing Workshop, pp.75-86, 2001.

[17] K. Ranganathan, A. Lamnitchi, I.Foster, " Improving data availability through dynamic model driven replication in large peer-to-peer communities", in CCGrid,p.376, 2002

[18] S. Park, J. Kim, Y. Ko, W. Yoon, Dynamic data replication strategy based on internet hierarchy BHR, Heidelberg: Lecture notes in computer science publisher, Springer-Verlag; pp. 838–846, 2004.

[19] P. Kunszt, E. Laure, H. Stockinger, K. Stockinger, "File-based replica management", Future Generation Computer Systems, vol. 22, no.1, pp.115-123, 2005.

[20] H. Jin, T. Cortes, R. Buyya, "High Performance Mass Storage and Parallel I/O: Technologies and Applications", IEEE Press and Wiley, 2002.

[21] J.M. Perez, F. Garcia, J. Carretero, A. Calderon, J. Fernandez, "A Parallel I/O Middleware to Integrate Heterogeneous Storage Resources on Grids (Grid Computing: First European Across Grids Conference, Santiago de Compostela, Spain, February 13-14, 2004), in: Lecture Notes in Computer Science Series, vol. 2970, pp. 124-131,2004.

[22] F. Garcia-Carballeira, J. Carretero, A. Calderon, J.D. Garcia, L.M. Sanchez, "A global and parallel file systems for grids", Future Generation Computer Systems, vol. 23, no 1, pp. 116-122, 2007.

[23] H. Zhong, Z. Zhang, X. Zhang. "A dynamic replica management strategy based on Data Grid", in Ninth International Conference on Grid and Cloud Computing, pp.18-23, 2010.

[24] M. Shorfuzzaman, P. Graham, R. Eskicioglu , "popularity driven dynamic replica placement in hierarchical data grid", in International conference on parallel and distributed computing, pp.524-531,2008.

[25] M. Shorfuzzaman, P. Graham, R. Eskicioglu, "Adaptive popularity-driven replica placement in hierarchical data grids", Journal of Supercomputers, vol. 51, pp. 374-92, 2010.

[26] M. Bsoul, A. Khasawneh, E. E. Abdallah, Y. Kilani, "Enhanced fast spread replication strategy for data grid", Journal of Network and Compute Applications, vol.34,pp. 575-80, 2011.