Middle-East Journal of Scientific Research 24 (11): 3501-3509, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.3501.3509

An Efficient Method for Replication of Data in Clouds Using Frequent Pattern Mining and Hybrid Cuckoo Search Algorithms

[1]M. Amutha and [2]R. Sugumar

[1]Department of CSE, PRIST University, Vallam, Thanjavur.- 613403, India
[2]Department of CSE, Velammal Institute of Technology, Panchetti, Chennai-601204, India

Abstract: Data replication strategy in recent years has evolved as an interesting research topic due to the availability of large quantity of data in the internet. The main aim of this research is to provide a better technique for solving the drawbacks that currently exist in the literary works of data replica method in cloud environment. Here, we intend to propose Data Replication system based on data mining techniques. Combination of replication algorithm and job scheduling policy for data replication in the data cloud environment through monitoring all job process. The data replication will be done by identifying the frequently used data patterns in the large database of a node. This will be done by frequent pattern mining algorithm where we incorporate HCS for optimization. The main contribution of the proposed method is popularity degree which is the access frequency based on time factor that is used in the first stage of the dynamic data replication strategy. In normal dynamic data replication strategy, the popularity degree is calculated considering the start time and the present time but in our modified dynamic data replication strategy, the popularity degree is calculated using access frequency based on time factor and access frequency based on users and the access frequency based on time factor is calculated using double exponential moving average function.

Key words: Replication · Pattern mining · Optimization

INTRODUCTION

Cloud computing is a large-scale parallel and distributed computing system. It consists of a collection of inter-connected and virtualized computing resources that are managed to be one or more unified computing resources. Further, the provided abstract, virtual resources, such as networks, servers, storage, applications and data, can be delivered as a service rather than a product. Services are delivered on demand to the end-users over high-speed Internet as three types of computing architecture, namely Software as a Service (SAAS), Platforms as a Service (PAAS) and Infrastructure as a service (IAAS). The main goal is to provide users with more flexible services in a transparent manner, cheaper, scalable, highly available and powerful computing resources [1]. The Software as a Service (SaaS) architecture provides software applications hosted and managed by a service provider for the end-user replacing locally-run applications with web services applications. In the Infrastructure as a Service (IaaS), Service includes provision of hardware and software for processing, data storage, networks and any required infrastructure for deployment of operating systems and applications which would normally be needed in a data center managed by the user. In the Platform as a Service (PaaS), Service includes programming languages and tools and an application delivery platform hosted by the service provider to support development and delivery of end-user applications [2].

Data cloud is to share and analyze the data resources, storage resources and others in a wide network which is dynamic, heterogeneous and distributive. The data distributed across a grid must be available and accessible to several applications with a reasonable performance [3]. It mainly focuses on analyzing massive data. In order to analyze the dynamic, real-time and online data, the whole data cloud system must be improve in order to enhance the access speed and reliability and safety and system's load balance. Therefore, how to choose the replication

Corresponding Author: R. Sugumar, Department of CSE, Velammal Institute of Technology, Panchetti, Chennai-601204, India

strategy for data cloud is particular important. Creating replica is to reduce access latency and bandwidth consumption, in other words, it is to reduce the average job execution time and improve the usage of cloud resources [4, 5].

In the data cloud system, it can be sure that, if there is not enough storage, a well-designed replication replacement algorithm will be needed. It can be gotten from the recent work, for example, the economy model has been used rather broadly and successfully in the data cloud research [6, 7]. The replication mechanism is divided into three important subjects: which file should be replicated, when to perform replication and where the new replicas should be placed. Usually, replication from the server to the client is triggered when the popularity of a file passes a threshold and the client site is chosen either randomly or by selecting the least loaded site [8].

However, as data sources and data processors integrated in a service application may be distributed geographically and connected with long-latency networks, data integration and sharing often lead to time and bandwidth penalties, thereby affecting the performance of the service application. This issue becomes more serious in large-scale, data intensive applications where large amounts of data have to be transported frequently between data sources and consumers. Data replication offers a practical solution to this issue by maintaining replicated copies of data in sites near to data consumers so as to reduce the time and bandwidth consumption of data transportation [9]. In addition, data replication can also improve the service performance and availability of data sources. Multiple replicated sites reduce the overhead imposed on a single point and if one replicated site is not available, users can have access to the copies on other nodes [10, 11].

Data replication can make the same data be store in different distributed sites. A good replication strategy can improve the performance of the grid environment, the stability of the grid environment and shorten the execution time of jobs. The bandwidth utilization is the most important factor to affect the overall downloading speed [12, 2]. The network environment is changeable, that makes the same replica sites are not always the best choices to download data to reduce the transmission time [13]. Replicas should be adjusted to the appropriate locations that are near to the computing devices to adapt the current network environment to reduce the time computing device to get the data and keep the environment in the higher performance [14].

Task Related: Tehmina Amjad *et al.* [15] discussed different issues involved in data replication were identified and different replication techniques were studied to find out which attributes are addressed in a given technique and which were ignored. A tabular representation of all those parameters is presented to facilitate the future comparison of dynamic replication techniques. The paper also includes some discussion about future work in this direction by identifying some open research problems.

Mohammad Bsoul *et al.* [16] proposed a dynamic replication strategy that was based on Fast Spread but superior to it in terms of total response time and total bandwidth consumption was proposed. This was achieved by storing only the important replicas on the storage of the node. The main idea of this strategy was using a threshold to determine if the requested replica needs to be copied to the node. The simulation results showed that the proposed strategy achieved better performance compared with Fast Spread with Least Recently Used (LRU) and Fast Spread with Least Frequently Used (LFU).

Zhe Wang *et al.* [17] proposed a dynamic data replication strategy based on two ideas. The first one employs historical access records which were useful for picking up a file to replicate. The second one was a proactive deletion method, which was applied to control the replica number to reach an optimal balance between the read access time and the write update overhead. A unified cost model was used as a means to measure and compare the performance of our data replication algorithm and other existing.

Dong Yuan *et al.* [18] proposed a matrix based k-means clustering strategy for data placement in scientific cloud workflows. The strategy contains two algorithms that grouped the existing datasets in k data centers during the workflow build-time stage and dynamically clusters newly generated datasets to the most appropriate data centers based on dependencies during the runtime stage. Simulations showed that the proposed algorithm can effectively reduce data movement during the workflow's execution.

Nazanin Saadat and Amir Masoud Rahmani [19] proposed a new dynamic data replication algorithm named PDDRA that optimizes the traditional algorithms. The proposed algorithm was based on an assumption: members in a VO (Virtual Organization) had similar interests in files. Based on this assumption and also file access history, PDDRA predicts future needs of grid sites and pre-fetches a sequence of files to the requester grid site, so the next time that this site needs a file, it will be locally available. This will considerably reduce access latency, response time and bandwidth consumption. PDDRA consists of three phases: storing file access patterns, requesting a file and performing replication and pre-fetching and replacement. The algorithm was tested using a grid simulator, OptorSim developed by European Data Grid projects. The simulation results showed that the proposed algorithm has better performance in comparison with other algorithms in terms of job execution time, effective network usage, total number of replications, hit ratio and percentage of storage filled.

Najme Mansouri and Gholam Hosein Dastghaibyfard [20] proposed a Dynamic Hierarchical Replication (DHR) algorithm that places replicas in appropriate sites i.e. best site that has the highest number of access for that particular replica. It also minimized access latency by selecting the best replica when various sites hold replicas. The proposed replica selection strategy selects the best replica location for the users' running jobs by considering the replica requests that waiting in the storage and data transfer time. The simulated results with Optor Sim, i.e. European Data Grid simulator showed that DHR strategy gives better performance compared to the other algorithms and prevents unnecessary creation of replica which leads to efficient storage usage.

Ming-Chang Leea *et al.* [21] proposed an adaptive data replication algorithm, called the Popular File Replicate First algorithm (PFRF for short), which had developed on a star-topology data grid with limited storage space based on aggregated information on previous file accesses. The PFRF periodically calculates file access popularity to track the variation of users' access behaviors and then replicates popular files to appropriate sites to adapt to the variation. Research had employed several types of file access behaviors, including Zipf-like, geometric and uniform distributions, to evaluate PFRF. The simulation results have showed that PFRF can effectively improve average job turnaround time, bandwidth consumption for data delivery and data availability as compared with those of the tested algorithms.

MATERIALS AND METHODS

Data Replication Strategy: Data replication, a prominent methodology from appropriated systems, is the major segment used as a part of the cloud for diminishing customer holding up time, extending data openness and minimizing cloud structure exchange speed use by offering the customer differing generations with a coherent state of similar organization. With the progress and evolution of various advancements, data replication and impersonation organization in spread structures have been thought about in various works, which are referenced and got in cloud data replication. Information replication calculations can be organized into two get-togethers: static replication and element replication calculations. The openness of record and detachment of the archive is found out to find the system byte powerful rate. The run of the mill dynamic replication method include three stages that are (i) which data record should be replicated and when to copy in the cloud structure, (ii) what quantity of duplication to be prepared in the cloud system and (iii) where this new duplicates are accumulated.

Proposed Technique for Data Replication: Three primary necessities of database replication are the execution, the accessibility and the consistency of information. These necessities are in struggle with each other in light of the fact that a change for the advantage of one of the paradigm infers a change (minimization) to the detriment of the other criteria. The entrance to a recreated element is commonly uniform with access to a solitary, non-reproduced element. The replication itself ought to be straightforward to an outer client. Moreover, in a disappointment situation, a failover of imitations is covered up however much as could reasonably be expected.

The main contribution of the proposed method is popularity degree which is the access frequency based on time factor that is used in the first stage of the dynamic data replication strategy. In normal dynamic data replication strategy, the popularity degree is calculated considering the start time and the present time but in our modified dynamic data replication strategy, the popularity degree is calculated using access frequency based on time factor and access frequency based on users and the access frequency based on time factor is calculated using double exponential moving average function. We also incorporate the synchronous and asynchronous updation for the newly created replicas. In synchronous updation, the record which we update in the main datacenter will get update simultaneously to the replicas in the sub datacenters but in asynchronous updation, the record which we update in the main datacenter will get update to the replicas after a specified time interval using the asynchronous agent. The Fig. 1 given below shows the flow diagram for the proposed method.

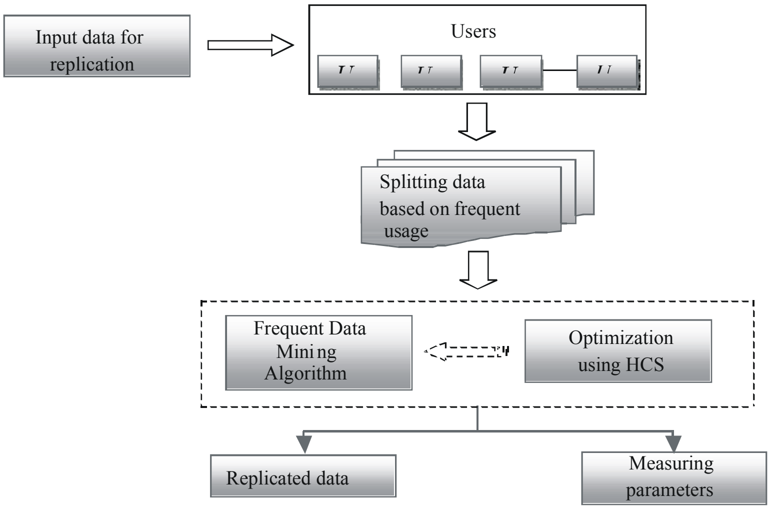
Input data for replication
Users
Splitting data based on frequent usage
Frequent Data Mining Algorithm
Optimization using HCS
Replicated data
Measuring parameters

Fig. 1: Block diagram for proposed method

As shown in the block diagram, the input data is collected and based on the frequent usage of particular datas the replication has to be created. The user performs various operation on certain datas on frequent times and those datas are required to be duplicated inorder for security and random usage which can save time and memory capacity. Creating replica is to reduce access latency and bandwidth consumption, in other words, it is to reduce the average job execution time and improve the usage of cloud resources. These best data replication algorithms are based on some kind's historical data access information and metadata. This information is in static condition. But, cloud is in dynamic environment. In cloud environment data replicate on a particular node. A data replica is best for some node at a certain point of time is not necessary to be best replication for another node at some different time. Because, workload, CPU capacity, changes in networks, etc. So, to select best data replication algorithm is still a problem. So inorder to perform the data replication in our proposed system we have employed frequent pattern mining for replication. The frequent pattern mining is a rule mining algorithm which can be utilized for effective data replication process due to its ease of operation and mining accuracy.

Frequent Pattern Mining Algorithm: Identifying frequent patterns is normally one of the principally huge ideas in information mining. A great deal of other information mining assignments and speculations come from this idea. The principle ventures in the successive example mining are to discover designs i.e. information which happen habitually in an information set. The continuous information mining can bolster the information replication procedure to a higher expand. The incessant example mining calculation is utilized as a part of our proposed technique since it gives enhanced mining rate and better precision is discovering definite copy in light of number of utilization of specific information in the dataset.

Let D_s be the data set for which the replication has to be obtained. Consider a set $R = \{a_1, a_2, \ldots\ldots, a_n\}$ which belong to the dataset D_s. The support value of the set R in the transaction database TR_d is the number of transaction in the cover of R in TR_d, which is given by the eqn,

$$S_t(R, Tr_d) = |C_t(R, TR_d| \tag{1}$$

where

$S_t()$ - Support of the item
$C_t()$ - Cover of the item

The frequency of the item set R in TR_d is the probability of R occurring in a transaction $Z \in TR_d$ which is given by the eqn 2 below,

$$F_q(R, TR_d) = F_q(R) = \frac{S_t(R, TR_d)}{|TR_d|} \tag{2}$$

An item set is called frequent if its support is not lower than absolute minimal support threshold ie. $0 \leq \lambda_{\min} \leq |TR_d|$. The collection of frequent item set in TR_d with respect to λ is given by the eqn 3 below,

$$Q_{col}(Tr_d, \lambda) = (R \subseteq P) S_t(R, TR_d) \geq \lambda \tag{3}$$

The threshold selection is done based on an optimization process where we use Hybrid cuckoo search for optimizing the values. This will aid in improving the mining scheme by finding out exact replica.

Optimization Using Hybrid Cuckoo Search (HCS): The cuckoo search algorithm is an optimization technique which is a biologically inspired algorithm designed based on the behavior of cuckoo. The cuckoo's behavior of laying egg is considered here for designing the algorithm steps. The ordinary cuckoo search is modified in our proposed technique by incorporating Gaussian function for calculating the fitness which is based on levy flight equation. The steps used in the Hybrid cuckoo search algorithm is explained in the below steps,

Step 1: Initialization Phase

The population ($p_{i,}$ where i=1, 2, n) of host nest is initiated arbitrarily.

Step 2: Generating New Cuckoo Phase

With the help of the levy flights a cuckoo is selected randomly which generates novel solutions. Subsequently, the engendered cuckoo is evaluated by employing the objective function for ascertaining the excellence of the solutions.

Step 3: Fitness Evaluation Phase

The fitness function is evaluated in accordance with Equations 4 and 5 shown hereunder, followed by the selection of the best one.

$$H_m = \frac{W_{Sel}}{W_{Tot}} \tag{4}$$

$$fitness = \max imum\ popularity\ = H_m \tag{5}$$

where,
W_{set} - signifies the selected population
W_{tot} - represents the total population

Step 4: Updation Phase

At the outset, the solution is optimized by the levy flights by employing the cosine transform. The quality of the novel solution is evaluated and a nest is selected arbitrarily from among them. If the quality of novel solution in the selected nest is superior to the previous solution, it is replaced by the novel solution (Cuckoo). Otherwise, the previous solution is treated as the best solution. The levy flights employed for the general cuckoo search algorithm is expressed by the Equation 6 shown below:

$$V_x{}' = V_x{}^{(k+1)} = V_x{}^{(k)} + \vartheta \oplus L_Y(c) \tag{6}$$

By suitably adapting Equation 6, levy flight equation using the Gaussian distribution is exhibited in Equation 7 here under:

$$V_x{}' = V_x{}^{(k+1)} = V_x{}^{(k)} + \vartheta \oplus \xi \tag{7}$$

where,

$$\xi = \xi_0 \exp(-\sigma C_g) \tag{8}$$

ξ_0,σ - represents the constants
C_g - Symbolizes the current generation

Step 5: Reject Worst Nest Phase

In this section, the worst nests are ignored, in accordance with their possibility values and novel ones are constructed. Subsequently, depending upon their fitness function the best solutions are ranked. Thereafter, the best solutions are detected and marked as optimal solutions.

Step 6: Stopping Criterion Phase

Till the achievement of the maximum iteration, the procedure is continued. Now the selected threshold is formulated for mining the frequent data using frequent pattern mining.

An association rule is an expression of the form $G \Rightarrow K$, then.

$$G \cap K = \{\} \tag{9}$$

where,
G and *K* are item sets.

Such a rule expresses the association that if a transaction contains all item in *G*, then that transaction also contains all item in *K*. Now *G* and *K* is called the body and the head of the rule.

From these various condition exist like the support rule of an association rule $G \Rightarrow K$ in TR_d, is the support of $G \cup K$ *in* TR_d and similarly the frequency of the rule is frequency of $G \cup K$. The association rule is called frequent if its frequency exceeds a given threshold value.

The accuracy of an association rule $G \cup K$ *in* TR_d is the conditional probability of having K contained in a transaction given that G is contained in that transaction by the expression given below,

$$C_p(G \Rightarrow K, TR_d) = \frac{S_t(G \cup K, TR_d)}{S_t(R, TR_d)} \tag{10}$$

Based on the above eqns the frequent pattern mining operates and the data replication is created which is then stored in the cloud nodes with security so that the users can use the frequently needed data at ease. The replicated data using the frequent pattern mining algorithm is better in terms of replication accuracy and the rate of occurrence. The record which we update in the main datacenter will get update to the replicas after a specified time interval using the frequent pattern mining algorithm.

RESULTS AND DISCUSSION

The performance of the proposed method is evaluated using synthetic datasets. Spatial database is used in the proposed method to store the information about the data replica. This information is kept in the form of synthetic datasets.

Performance Analysis: In our proposed work, frequent pattern mining algorithm is used for data replication process. In order to analyze the performance of the proposed work, we have measured various parameters based on the execution scenario. The table 1 given below shows the data replica obtained using our proposed method for different threshold values,

The Fig. 2 given below shows the graphical representation for the above measures. For various threshold value the replica number are plotted and is represented in the form of bar chart.

The Table 2 given below shows the response time and replica number obtained using our proposed system for varying task load. For different task load percentage , the response time is calculated and the number of replication is also noted down.

Table 1: Replica number for different threshold

No	Threshold for frequent item	Replica number
1	0.1	22
2	0.15	12
3	0.2	9
4	0.25	7

Table 2: Response time and replica number for varying task load

Task load (%)	Response time (s)	Replica number
10	154631	5
20	165487	10
30	176542	15
40	189621	20

Table 3: Response time and Replica number for proposed and existing method

Methods	Response time (s)	Replica number
Proposed Frequency pattern mining	171570	12.5
Association rule mining	202560	8

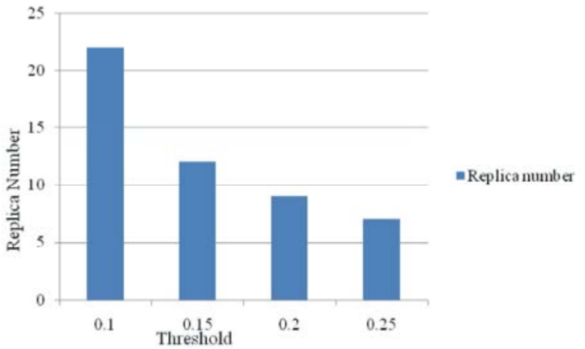

Fig. 2:

The Fig. 3 and Fig. 4 shows the graphical representation of response time versus task load and replica number respectively. From the graph it is clear that the replica number increases as the number of task increases which prove that the replication process carried out using the proposed technique is efficient in terms of the replication ratio.

The proposed system of data replication using frequent pattern mining is compared with the association rule mining to prove the effectiveness of the proposed system. The Table 3 given below shows the measures obtained using the proposed and existing method.

The Fig. 5 and Fig. 6 gives the comparative chart for proposed and existing method. From the graph it is clear that our proposed system has better response time and replica number when compared with that of the existing system.

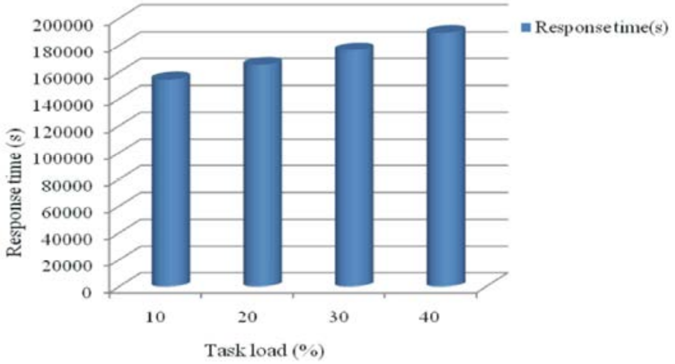

Fig. 3: Graphical representation of task load versus Response time

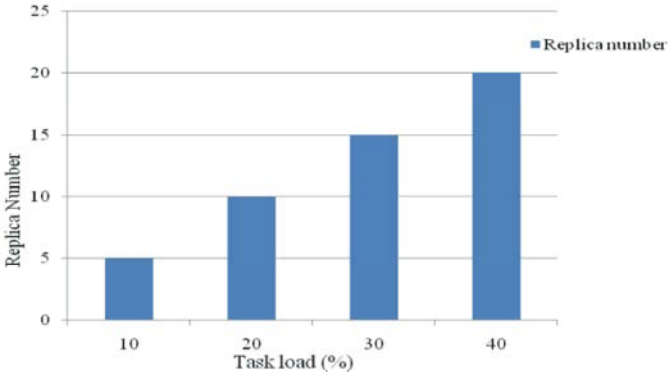

Fig. 4: Graphical representation of task load versus Replica number

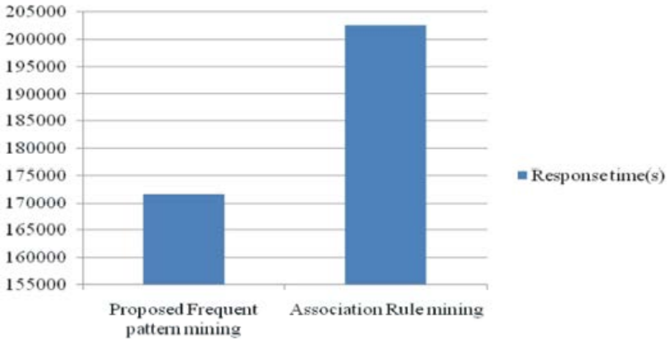

Fig. 5: Graphical representation of comparison for Response time using proposed and existing method

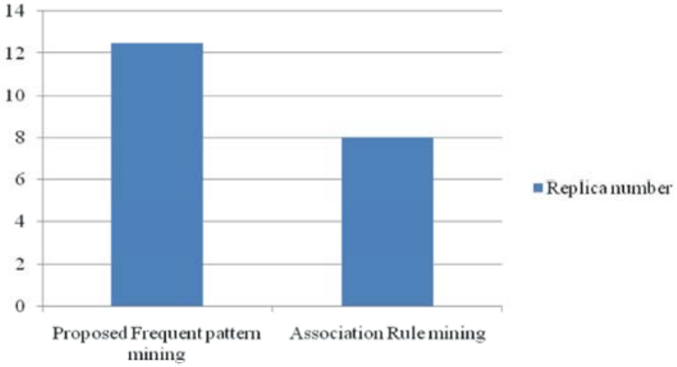

Fig. 6: Graphical representation of comparison for Replica number using proposed and existing method

CONCLUSION

In this paper we h ave proposed an efficient method for data replication using frequent pattern mining. The data replication will be done by identifying the frequently used data patterns in the large database of a node. This will be done by frequent pattern mining algorithm. The main contribution of the proposed method is popularity degree which is the access frequency based on time factor that is used in the first stage of the dynamic data replication strategy. From the results it is clear that the proposed system of data replication ensures better response time and replication number when compared with existing system. In future we plan to incorporate different mining algorithm to refine the response time further.

REFERENCES

1. Tehmina Amjad, Muhammad Sher and Ali Daud, 2012. “A survey of dynamic replication strategies for improving data availability in data grids”, Future Generation Computer Systems, 28: 337-349.
2. Mohammad Bsoul, Ahmad Al-Khasawneh, Yousef Kilani and Ibrahim Obeidat, 2012. “A threshold-based dynamic data replication strategy”, The Journal of Supercomputing, 60(3): 301-310.
3. Zhe Wang, Tao Li, Naixue Xiong and Yi Pan, 2012. “A novel dynamic network data replication scheme based on historical access record and proactive deletion”, The Journal of Supercomputing, 62(1): 227-250.
4. Dong Yuan, Yun Yang, Xiao Liu and Jinjun Chen, 2010. “A data placement strategy in scientific cloud workflows”, Future Generation Computer Systems, 26(8): 1200-1214.
5. Nazanin Saadat and Amir Masoud Rahmani, 2012. “PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids”, Future Generation Computer Systems, 28(4): 666-681.
6. Najme Mansouri and Gholam Hosein Dastghaibyfard, 2012. “A dynamic replica management strategy in data grid”, Journal of Network and Computer Applications, 35(4): 1297-1303.
7. Ming-Chang Leea, Fang-Yie Leub and Ying-ping Chen, 2012. “PFRF: An adaptive data replication algorithm based on star-topology data grids”, Future Generation Computer Systems, 28(7): 1045-1057.
8. Ruay-Shiung Chang, Hui-Ping Chang and Yun-Ting Wang, 2008. “A Dynamic Weighted Data Replication Strategy in Data Grids”, In Proceeding of IEEE/ACS International Conference on Computer Systems and Applications, pp: 414- 421.
9. Chao-Tung Yang, Chun-Pin Fu and Chien-Jung Huang, 2007. “A Dynamic File Replication Strategy in Data Grids”, In Proceeding of IEEE/ACS International Conference on TENCON, pp: 1-5.
10. Stockinger, H., 2001. “Database Replication in Worldwide Distributed Data Grids”, University of Vienna.
11. Xiaohua Dong, Ji Li, Zhongfu Wu, Dacheng Zhang and Jie Xu, 2008. “On Dynamic Replication Strategies in Data Service Grids”, In Proceeding of 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), pp: 155-161.
12. Ruay-Shiung Chang, Hui-Ping Chang, 2008. “A dynamic data replication strategy using access-weights in data grids”, Journal of Supercompute.
13. William H.Bell, David G. Cameron, Ruben Carvajal-Schiaffino, A. Paul Millar, Kurt Stockinger and Floriano Zini, 2003. “Evaluation of an Economy-Based File Replication Strategy for a Data Grid”, In Proceedings of the 3rd International Workshop on Agent based Cluster and Grid Computing at International Symposium on Cluster Computing and the Grid, IEEE Computer Society, pp: 661-668.
14. Mark Carman, Floriano Zini, Luciano Serafini and Kurt Stockinger, 2002. “Towards an Economy-Based Optimization of File Access and Replication on a Data Grid”, In Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Computer Society, pp: 340-345.
15. Faouzi Ben Charrada, Habib Ounelli, Hanène Chettaoui, 2010. “An Efficient Replication Strategy for Dynamic Data Grids”, In Proceeding of IEEE/ACS International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), pp: 50-54.
16. Gao Gai-Mei and Bai Shang-Wang, 2012. “Design and Simulation of Dynamic Replication Strategy for the Data Grid”, In Proceeding of IEEE/ACS International Conference on Industrial Control and Electronics Engineering (ICICEE).
17. Feng Dan and Tan Zhipeng, 2006. “Dynamic replication strategies for object storage systems”, In Proceeding of the 2006 international conference on Emerging Directions in Embedded and Ubiquitous Computing, pp: 53-61.

18. Isa, A.M., A.N.M.M. Rose, M. Mat Deris and M. Zarina, 2010. "Dynamic data replication strategy based on federation data grid systems", In proceeding of the First international conference on Information computing and applications, pp: 25-32.
19. Ming Lei, Susan V. Vrbsky and Xiaoyan Hong, 2006. "A Dynamic Data Grid Replication Strategy to Minimize the Data Missed", In Proceeding of Broadband Communications, Networks and Systems, pp: 1-10.
20. Wenhao Li, Yun Yang, Dong Yuan, 2011. "A Novel Cost-Effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres", In Proceeding of IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing.
21. Rajkumar Buyyaa, Chee Shin Yeoa, Srikumar Venugopala, James Broberga, Ivona Brandicc, 2009. "Cloud computing and emerging IT platforms: Vision, hype and reality for delivering computing as the 5th utility", Future Generation Computer Systems, 25(6): 599-616.
22. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia, 2010. "A view of cloud computing", Communications of the ACM, 53(4): 50-58.
23. Mohamed K. Hussein and Mohamed H. Mousa, 2012. "A Light-weight Data Replication for Cloud Data Centers Environment", International Journal of Engineering and Innovative Technology (IJEIT), 1(6).
24. Bakhta Meroufel and Ghalem Belalem, 2012. "Dynamic Replication Based on Availability and Popularity in the Presence of Failures", Journal of Information Processing Systems, 8(2).