IUST Iran University of Science and Technology
Iranian Journal of Electrical and Electronic Engineering
Journal Homepage: ijeee.iust.ac.ir

Improving Data Availability Using Combined Replication Strategy in Cloud Environment

N. Mansouri*(C.A.) and M. M. Javidi*

Abstract: As grow as the data-intensive applications in cloud computing day after day, data popularity in this environment becomes critical and important. Hence to improve data availability and efficient accesses to popular data, replication algorithms are now widely used in distributed systems. However, most of them only replicate the static number of replicas on some requested chosen sites and it is obviously not enough for more reasonable performance. In addition, the failure of request is one of the most common issue within the data centers. To compensate these problems, we, propose a new data replication strategy to provide cost-effective availability, minimize the response time of applications and make load balancing for cloud storage. The proposed replication strategy has three different steps which are the identification of data file to replicate, placing new replicas, and replacing replicas. In the first step, it finds the most requested files for replication. In the second step, it selects the best site by consideration of the frequency of requests for replica, the last time the replica was requested, failure probability, centrality factor and storage usage) for storing new replica to reduce access time. In the third step, the replacement decision is made in order to provide better resource usage. The proposed strategy can ascertain the importance of valuable replicas based on the number of accesses in future, the availability of the file, the last time the replica was requested, and size of replica. Our proposed algorithm evaluated by CloudSim simulator and results confirmed the better performance of hybrid replication strategy in terms of mean response time, effective network usages, replication frequency, degree of imbalance, and number of communications.

Keywords: Data Replication, Cloud Computing, CloudSim, Replica Placement.

1 Introduction

UNIQUE properties of Cloud computing in data storage with acceptable efficiency as well as low cost, enhanced the accessible capability of data over the Internet. This approach enable the user to decrease the load of local data storage, increase the security and allows to the professional users to share the data flexibly. Features of next-generation scientific study, the e-Science needs imposed by these characteristics, and main enabling technologies are presented in Fig. 1. Fig. 1 presents that web services, workflow, Semantic web, Grid computing, Cloud computing (e.g. SaaS, PaaS, IaaS), etc. are some of major enabling digital facilities for presenting useful e-Infrastructure and application-oriented platforms. As a new computing system, it presents main challenges and opportunities involving technical, cultural and business problems [1].

Generally, the cloud system presents the infrastructure for software and hardware as services by several data centers. Consequently, cloud system transfers the computation and storage from the consumers onto servers of data centers. Therefore, it is challenging to present efficient and fast access to the data center of cloud because of the large scale, distributed, and dynamic nature of the cloud computing. In different fields such as high energy physics and metallurgy the volume of data is already measured in petabytes or terabytes and large data sets are emerging as critical community resources. Data replication is one of the

Iranian Journal of Electrical and Electronic Engineering, 2019.
Paper first received 21 November 2017 and accepted 06 October 2018.
* The authors are with the Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran.
E-mails: najme.mansouri@gmail.com and javidi@uk.ac.ir.
Corresponding Author: N. Mansouri.

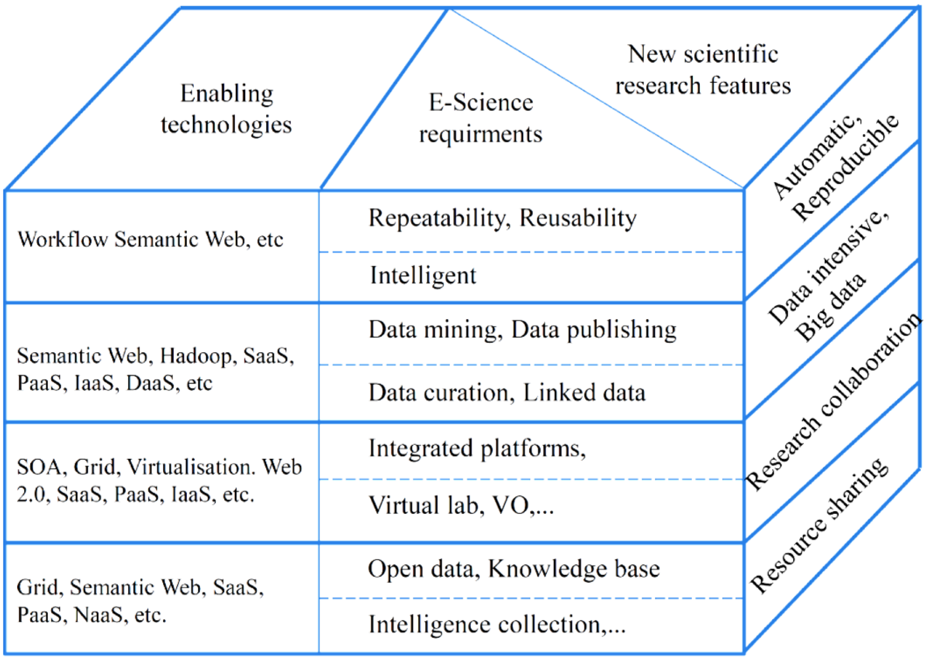

Fig. 1 A summary of e-Science requirements and main technologies [1].

most common techniques to enhance the availability and reliability of data storage service in Grid and Cloud computing [2-7]. Data replication provides the possibility of large-scale parallel read/query management in cloud environment by creation of multiple data replicas and distributed them through the various cloud nodes [8]. Moreover, data replication is able to decrease the user waiting time, improve the availability of data and decrease the cloud bandwidth consumption by suggestion of various copies from specific service on different nodes. However, for successful data replication some consideration must be regards:

- The amount of data replicas: In spite of replication advantageous; excessive replication can be increased the complexity of data consistency and caused to overconsumption of storage space. Therefore, professional satisfaction of data replication parameters include of the saving of earlier replicas when the data hotspot appears, deleting the redundant replicas when the hotspot cools down and permission of the request traffic fluctuation to avoid unnecessary replicas are so beneficially.
- The distribution of data replicas: it consider the replica storage, query, and update costs. In a dynamic cloud system, data may be saved in various cloud hosts for higher query load and performances. The conventional cloud storage system, e.g., Amazon's Dynamo [9], locate the replicas at a constant number of physical nodes. While, the others do not consider the geographical diversity, access cost, and replication cost issues [10]. Consequently, a high access cost, and unbalance workload may be evolved.

Therefore, the replica placement has administrated role in network usage as well as access skew, especially for a large-scale inhomogeneous storage system in which each data node have special capabilities. The most disadvantageous of random replica placement strategy are load imbalance across the cloud storage as well as poor parallelism and low performance. To compensate these weaknesses, authorized response must be provide for the appropriate threshold for replica placement in a way that distribute the workload on node of cluster. To address this issue an adaptive replica placement strategy, which stores the replicas in the suitable location, must be designed in a balanced way. In this paper, we propose a new data replication strategy called Combined Replication Strategy (CRS) that selects the best site for storing new replica based on number of access, the last time the replica was requested, failure probability, centrality factor and storage usage to reduce access time. Since, the replica placement consumes the storage capacity of each node, the presence of an efficient replica replacement strategy is necessary to improve the overall performance. To solve the storage restriction, a new replica replacement algorithm with the ability to delete a replica with the minimum favorability in the future has been proposed. In order to evaluate proposed CRS strategy, it is implemented in CloudSim simulator [11] and the simulation results confirmed the significant important, especially in response time, effective network usages, respect to the other presented strategies.

At following in Section 2, the abilities of cloud computing within the distributed systems are explained. Section 3 gives a survey on the literatures about the data replication algorithms. Section 4 illustrated the system

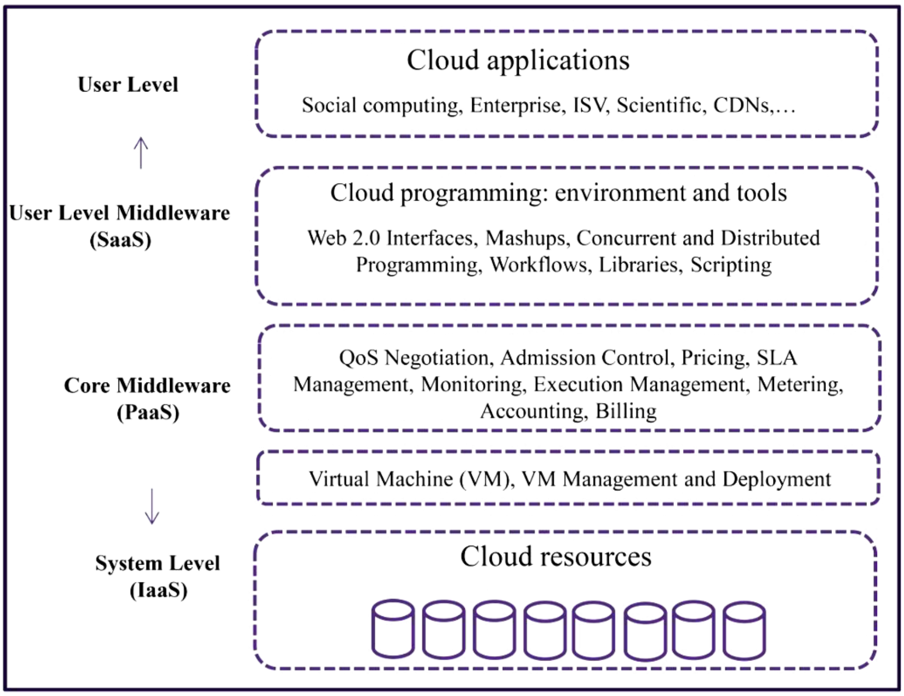

Fig. 2 Layered cloud computing structure [11].

model in details. The details of proposed strategy are explained in Section 5. Section 6 illustrates the results of simulation. Final section summarizes the major contributions, research results and the future work.

2 Cloud Computing Versus Other Large-Scale Distributed Computing

Cloud computing i.e., a novel pattern of business computing, emerges from development of grid environment, parallel computing and distributed computing. Layered structure of cloud computing is presented in Fig. 2. It consists of three main layers as following:

User-Level Middleware: User level includes different tools like Web 2.0 Interfaces to provide appropriate user-interfaces. There are several programming toolkits to develop and run applications of cloud. In addition, it consists of frameworks to provide multi-layer applications development, like Spring and Hibernate.

Core Middleware: Core layer consists of implementation for platform-level services to provide run-time environment for maintaining and managing application of previous layer. For example, there are Dynamic SLA Management, Accounting, Billing, Execution monitoring and management, and Pricing in this layer. The main services operating of Core layer are Amazon EC2, Google App Engine, and Aneka.

System Level: A set of data centers that are commonly installed with various hosts constructs the computing capability of cloud system. The higher-level virtualization technique and toolkits control their servers and share their resources among virtual instances of servers.

In comparison with grid computing; cloud computing has some unique specifications; (i) grid computing is the integration of inhomogeneous distribution resources while, the cloud computing applied in system with huge concentrated data center resources. Moreover, virtualization manner do not show the inhomogeneity of the resources in cloud computing, (ii) application of grid is convenient in science computation, especially solving the particular problem; while the cloud computing is most popular to satisfy the various requirement of users, and (iii) the resources are packed into virtual resources using virtualization technique in cloud computing.

Task scheduling to resource pooling which organizes by different computers is a fundamental process of cloud system. In addition, the other interesting properties of cloud are commercialization and the virtualization techniques [12-13]. For instance, it transfers complexity of job assignment to the virtual machine layer based on the resource virtualization procedure. The main differences between large-scale distributed and cloud systems are listed as follows [14-15].

On-Demand Self-Service: A user is able to one-sidedly provision computing power, like network storage without different interactions with provider of services.

Broad Network Access: All Capabilities are provided over the network and employed by predefined strategies that provide the use by inhomogeneity thin or thick client platforms

Resource Pooling: The computing resources are available to satisfy requirement of users based on the multi-tenant scheme in various physical and virtual resources.

Rapid Elasticity: Abilities can be elastically provisioned as quick as possible and automatically, to increase the rate of scale out as well as rate of scale in. In view of users, the abilities are available for provisioning to be unlimited and can be getting in any time and quantity.

Measured Services: Cloud systems in automatic way manipulate and control resource usage by leveraging a metering ability at some levels of abstraction based on the service type. It is possible to manage resource usage, and reported to present transparency for both the and users.

Briefly, the most outstanding characteristics of clouds are high availability, reliability, lower cost, high flexibility and scalability computing environment in comparison to the other large-scale distributed computing systems [16-20].

3 Related Works

Data replication is one of the most common methodologies in cloud environment to decrease the waiting time as well as bandwidth consumption by creation of replicas in the most appropriate site and consequently enhance the data availability [21]. Since, data replication and replica management is a hot issue for researchers in distributed systems.

Wei *et al*. [22] presented a cost-effective dynamic replication management scheme referred as CDRM. Firstly, they model the availability and replica number relationship and set the number of replication in the lowest possible value to satisfy the user availability requirement. If the replica number is lower than the proposed threshold by model, more replicas must be generated in data nodes of cluster. CDRM used from these characteristics as well as blocking probability of data nodes as criteria for replica placement. The authors introduced CDRM with HDFS and concluded that CDRM is able to enhance the performance and load balancing of HDFS as default.

Hussein *et al*. [23] illustrated an Adaptive Replication Strategy (ARS) in the cloud environment. They studied the availability and file accessibility as affecting parameters to enhance the data file reliability on the base of estimation of the user access to block of each file. Moreover, ARS extends the replication of large-scale various file on different sites by minimization of cost according to the heuristic method. ARS is able to determine the most popular file on the base of evaluation in the recent history of file data access using HLES times series. When, replication factor (i.e., estimation on the base of popularity of file) is lower than specific value, the replication is done. Therefore, employments of heuristic method enables ARS to determine the best threshold during replication process. Simulation results confirm the adaptive strategy improves availability in cloud environment. Consideration of one parameter, i.e., the popularity degree, replica placement is the most restriction of proposed method.

Rajalakshmi *et al*. [24] presented a Dynamic Replica Selection and Placement (DRSP) to enhance the availability of data in the cloud. DRSP consider file application and replication operation as two major processes. The former includes of replica location and creation using index and catalog elements. The latter process performs by getting the information of sites in each data. Using of index enables DRSP to store the replica in appropriate location (local/ remote) by maintenance of master and slave. Selection is done based on the ratio of replication threshold to the total number of request. Eucalyptus cloud environment used for evaluation and the results confirmed the ability of DRSP in reducing bandwidth consumption and response time.

Hussein *et al*. [25] illustrated a Light-weight Data Replication (LWDR) strategy and improved the access time in cloud. LWDR is able to estimate the necessary number of replicas and the most appropriate node for placement. A pre-estimation threshold based on the availability of each existing replica is calculated as replication factor criterion. The number of replicas changed adaptively to enhance the total availability factor. To estimate the popularity degree of data, Holt's Linear and Exponential Smoothing (HLSE), i.e., a facile computationally times series prediction method is employed. HLES can smooth and provide short-term predictions of arrival rates of requests and demand rates of service. Accordingly, the utilization of each server host and the future response time for service are calculated and the possibility of future response time of web service is provided. Neglecting of failure probability as well as reliability factors are the most shortcomings of LWDR.

Sun *et al*. [26] used the analytically method to determine the system availability and number of replicas relationship by consideration of the size, access time and failure probability of each data file as criteria. The proposed method determines the popularity data file using analysis of access history and definition of suitable weights for different access data. Typically, the more recently access data is a good candidate for achieving a large weight. Replication process will be ended if the popularity of a file is higher than dynamic predefined value which is obtained based on the number of access by all users. Validation of proposed method is carried out by CloudSim toolkit and confirmed the ability of proposed strategy for improvement of system availability as well as enhancement of system task successful execution rate.

Chen *et al*. [27] presented an adaptive data replication management strategy, that manages the data replication process on the base of dynamic-window statistical analysis of the response time for requesting data. It is able to provide higher workload balance as well as efficiently than the other proposed request-oriented methods, that use the request traffic as criterion for data replication and the random method that uses the constant number of replicas and static data replication strategy. The proposed method is especially preferred for active data-center (ADC) structures, that encapsulates independent data unit to improve self-protection and self-defense characteristics. This type of data structure revealed higher performance than the conventional one due to its compression characteristics.

Zhang *et al*. [28] proposed a new replica placement strategy using bidding mode [6] in a cloud storage system. They integrated the bidding mode with replica technology. If the file availability did not satisfy the user requirements, the bidding strategy is employed to create new replica in the most appropriate location. The adaptability of proposed strategy to changes of environment is achieved by maintenance of rational replicas. This characteristic provides reasonable availability level, reduces access latency, and enhances load balancing. Neglecting of size of storage is the most limitation of this replication algorithm.

Mansouri *et al*. [29] proposed a Utility-base Data Replication (UDR) algorithm that enhances file retrieval time. The main contribution of UDR is replica selection procedure based on the storage access latency, distance between nodes, and network performance. UDR is able to store the replica in the most suitable site that the file has been accessed for the most time instead of storing files in many sites. They compared UDR with LRU, LFU, BHR, DHR, LALW, MDHRA and PHFS based on the various file access patterns by OptorSim simulator. The experiments demonstrated that UDR algorithm outperforms the other algorithms in total job time and effective network usage. But, the most shortcoming of UDR is where it does not address failure probability in replica placement. It was necessary to note the failure of data center is very common rather than exceptional.

Kirubakaran *et al*. [30] proposed a Modified Dynamic Data Replication Strategy (MDDRS) based on the synchronous and asynchronous process. In the first step, MDDRS calculates the popularity value based on double exponential moving average relation. In the second step, determines the necessary number of replicas. In the third step, it incorporates the synchronous and asynchronous updating for new replicas. Synchronous updating process scatters the modifications of replicas in main data center to sub datacenters. While asynchronous process sends the modifications of main datacenter after a predefined time interval. The simulation results demonstrated that MDDRS enhanced the system availability since it replicated more recently accessed file. The main drawback of MDDRS is that it did not consider the storage restriction and replacement strategy.

We notice that there is only few number of works that consider the heterogeneous nature of the environment, especially in cloud. Moreover, not all work focuses on failure probability and centrality factor in replica placement. This motivates us to develop a dynamic replication strategy for heterogeneous cloud systems focusing on failure probability along with maintaining higher data availability and better performance.

4 System Model

Usually clouds with its unique characteristics provide the elasticity as well as scalability for user by employment of large and power-consuming data centers and usable strategies in distributed storage system, e.g., GFG [31], HDFS [32].

CRS considers the several clusters that consist of m independent heterogeneous nodes *D1*, ..., *Dj*, ..., *Dm* to store *n* different data files *f1*, ..., *f2*, ..., *fn*. Fig. 3 schematically shows the proposed structure for distribution of n files into *m* nodes. In this work, we suppose that the access pattern to file fi is sequentially as the same as most system file [33-34]. Also, file partitioning is neglected and consequently entire file is placed on to one storage. This assumption does not restrict the popularity of presented strategy and every file segment treats as a stand-alone file. Also, due to read only characteristics of all the data, no data consistency procedure is required.

5 Combined Replication Strategy (CRS)

If the popular files define as the data file that one more needed, then it is possible to decrease the data access and job execution time by introduction of appropriate strategy to find and replicate the popular file in required sites. Our proposed methodology contains two subsections as replica placement and replica replacement.

5.1 Replica Placement

The presence of some replicas from each file is an important parameter for improvement of performance in cloud environment. Because placing several replicas of

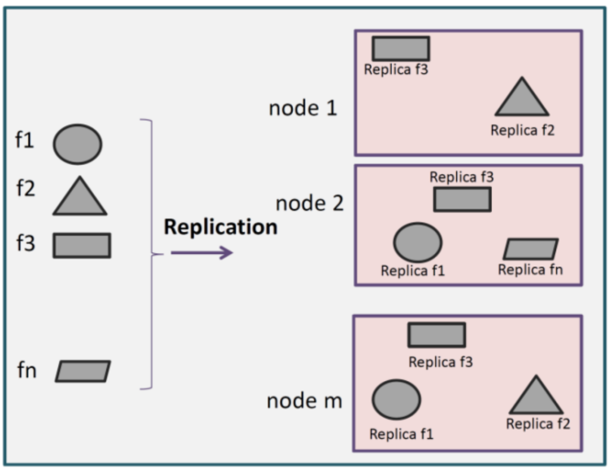

Fig. 3 Schematically representation of system model for replication management.

one file in one storage, not only do not help to improve data availability but also wastes huge amount of resources. One of the serious concerns in cloud with high-speed growth of data is finding the best site for replica placement.

When necessary file does not available in the local site, the file replication will be triggered. In this condition CRS, firstly determines the best storage element (*BSE*) using the maximum value of SE (*VSE*) as threshold. *VSE* is estimated by consideration of the frequency of requests of replica, the last time the replica was requested, storage usage (*SU*), and failure probability (*FP*). Combination of these parameters is able to determine the possibility of needed replica, again. According to the temporal locality (recently accessed file are likely to be accessed again), number of replica access has a main role in replica placement decision [35]. Moreover, the relative importance of a site in the system can be determined by the centrality of a node in a graph. Our strategy considers the centrality to reduce retrieval time. There are different centrality metrics such as closeness centrality, degree centrality, between centrality, and eccentricity centrality [36]. We only consider the closeness metric in replica placement process. A site is set as closeness in a network, if it has the lowest value for the summation of the distances from all of the other sites. The lower the sum of distances from the other sites, the more centrality has the site. The closeness centrality value for site v can be defined as the following [36]:

$$Centrality(v) = \frac{N-1}{\sum_{a \neq b} d(a,b)} \tag{1}$$

The parameter N is used to indicate total number of sites in the system and $d(a,b)$ shows the distance between site a and site b.

$$VSE = Centrality + FR_i + \frac{1}{SU} + \frac{1}{FP} + \frac{1}{CT - LT_i} \tag{2}$$

In which CT is the current time, LT_i is the last request time of replica i, and FR_i is the frequency of requests of the replica i. Note that, these five parameters are normalized to [0,1].

5.2 Replica Replacement

When sufficient space for new replica does not available, some replicas should be candidate for deletion based on the CM value:

$$CM = \frac{N}{S} + \frac{1}{CT - LT} + \frac{1}{P} \tag{3}$$

In which N is the probability of file access number for each file in the future and estimation using exponential growth/decay. S is the size of particular replica. CT is the current time, LT is the last request time of particular replica, P is data availability. In replica deletion step, it arranges all of files in the best site according to the *CM* value in ascending order. More explanation of these parameters are described:

Availability (P): We consider the data availability for each storage site that shows the probability of existing particular file in it. If availability of all stored files in one storage are same, then the file availability in the storage element j is shown by P_{SEj}.

In addition, we know more than one replica of file may be stored, thus the availability of each f_i file that (P_i) will be obtained by [37]:

$$P_i = 1 - \prod_{j=1}^{N} (1 - P_{SEi}) \tag{4}$$

where N indicates the number of f_i files copies. The probability of unavailability is obtained form (1-P_i), of course with this assumption that the access operation of files will done independently.

Number of access in future (N): The exponential growth/decay is employed to estimate the number of access file in future. This function explains various actual phenomena, e.g. bacteria, radioactive, as a function that shows how things grow or decay during the time. This model can be used in access history as well, since each file has number of access that increases by the increase of access rate and vice versa. The process of accessing files in cloud environment obeys an exponential model. If n_0 is the number of access for the file f at time t, and $n(t)$ is the number of access for the same file at time t+1 (just after the first access). The exponential decay/growth model can be depicted as:

$$n(t) = n_0 \times e^{-rt} \tag{5}$$

Suppose T is the number of intervals transferred, F is the file set that have been requested and n_f^t represents the number of access for the file f at time interval t, and then the sequence of the access numbers is estimated by:

$$n_f^0 \; n_f^1 \; n_f^2 \cdots n_f^{T-1} \; n_f^T \tag{6}$$

Thus, using the exponential decay/growth model:

$$n_f^T = n_f^{T-1} \times e_{T-1}^{\alpha} \tag{7}$$

This implies that:

$$\alpha_{T-1} = \ln\left(\frac{n_f^T}{n_f^{T-1}}\right) \tag{8}$$

So, the average rate for all intervals is:

$$\alpha = \sum_{i=0}^{T-1} \frac{\alpha_i}{T} \tag{9}$$

And it is possible to determine the number of access for next time interval:

$$\alpha_f^{T+1} = \alpha_f^T \times e^{\alpha} \tag{10}$$

For example, if 23, 20, 12, 10 are the number of access for file *A* during four intervals, respectively. Then firstly the average decay/growth rate for file *A* is computed as:

$$\alpha = \frac{1}{3}\left[\ln\left(\frac{20}{23}\right) + \ln\left(\frac{10}{20}\right) + \ln\left(\frac{10}{12}\right)\right] = -0.27 \tag{11}$$

Then, it estimates the next number of access for file *A* as:

$$\alpha_A^5 = 10 \times e^{-0.27} = 7.6 \approx 8 \tag{12}$$

The replacement will be done while, the value for storing new copy is higher than the aggregation value of candidate replicas for deletion. The value of replicating f_i file is estimated by:

$$\Delta p_i = (P_i' - P_i) \times N_i \tag{13}$$

That P_i is the present availability of f_i file and P'_i is the availability of f_i file after replication. Also, the expense of deleting the candidate files will be estimated by the formula:

$$\sum_{j \in Candidates} (P_j' - P_j) \times N_j \tag{14}$$

In this formula, the present availability for candidate file is shown by P_j and the availability for the candidate file after deletion is indicated by P'_j. If value of replicating f_i *is* greater than the accumulative value loss in the candidates file deletion from the best site, then file f_i replicated in the best site. Where

$$\Delta p_i N_i > \sum_{j \in Candidates} (P_j' - P_j) \times N_j \tag{15}$$

Fig. 4 describes the replacement strategy.

6 Simulation and Performance Evaluation

This section presents CloudSim architecture and discusses the simulation results as well as their discussion.

```
Create list L from files in BSE and for each file compute CM (Eq. 3);
Sort list L in ascending order of CM;
Sum = 0;
While (L is not empty && not enough space for new replica) {
        Select file f_i from list L;
        Sum = Sum + (p_i' − p_i) × N_i;
}
If (Δp_F × N_F > Sum) { delete candidates files from list L in BSE;}
If (enough space exist for new replica of F in BSE) { store new replica of F;}
```

Fig. 4 Replacement strategy.

6.1 CloudSim Architecture

Fig. 5 schematically represents multi-layered design of the CloudSim architectural elements. Initial version of CloudSim employs SimJava as the discrete event simulation engine [38]. SimJava is equipped with various functionalities, e.g., queuing of events, generation of cloud system entities (services, host, data center, broker, VMs), their interactions and management of modeling clock. In present version of CloudSim, the SimJava layer has been omitted to let some novel characteristics. CloudSim modeling layer is able to model virtualized management interface for virtual machines, memory, storage, and bandwidth. This layer has main functions such as provisioning of hosts to virtual machines. A host may be allocated to the several virtual machines that run different programs according to the SaaS provider's determined QoS levels. Application designer can extend functionalities of this layer to evaluate application performance.

The first layer of Cloudsim is the User Code that provides the key elements, e.g., hosts, tasks with requirements, virtual machines, number of users, and task scheduling strategies. Then, application developer can do some operations such as: (i) define different workload configurations; (ii) model and test cloud availability by particular setting; and (iii) implement custom application provisioning approaches based on the their federation.

6.2 Experimental Settings

In our case, 64 data centers are geographically distributed in cloud (Fig. 6). The service providers illustrated by 1000 virtual machines (VMs). Each VM is configured as 512MB RAM, 1000 MIPS, 1000M bandwidth, and single processer. One hundred various data files are situated in the cloud storage environment, with each size in the range of [0.1, 10] GB. 1000 jobs are submitted to the service providers based on the Poisson distribution. Each task needs 1 or 2 data files, randomly. The host allocates query requests to each VM by the VMSchedulerTimeShared policy. At first suppose, the number of replicas of each data file equal 1 and are placed randomly.

7 Simulation Results and Analysis

Average Response Time (*ART*) and Effective Network Usage (ENU), Replication Frequency (RF), Degree of Imbalance (DI), and (Number of Communications (NC) are considered as the performance evaluation metrics.

7.1 Average Response Time (ART)

ART: It estimates by averaging of response time, i.e., the interval between the sending and returning of request, of all data using (16).

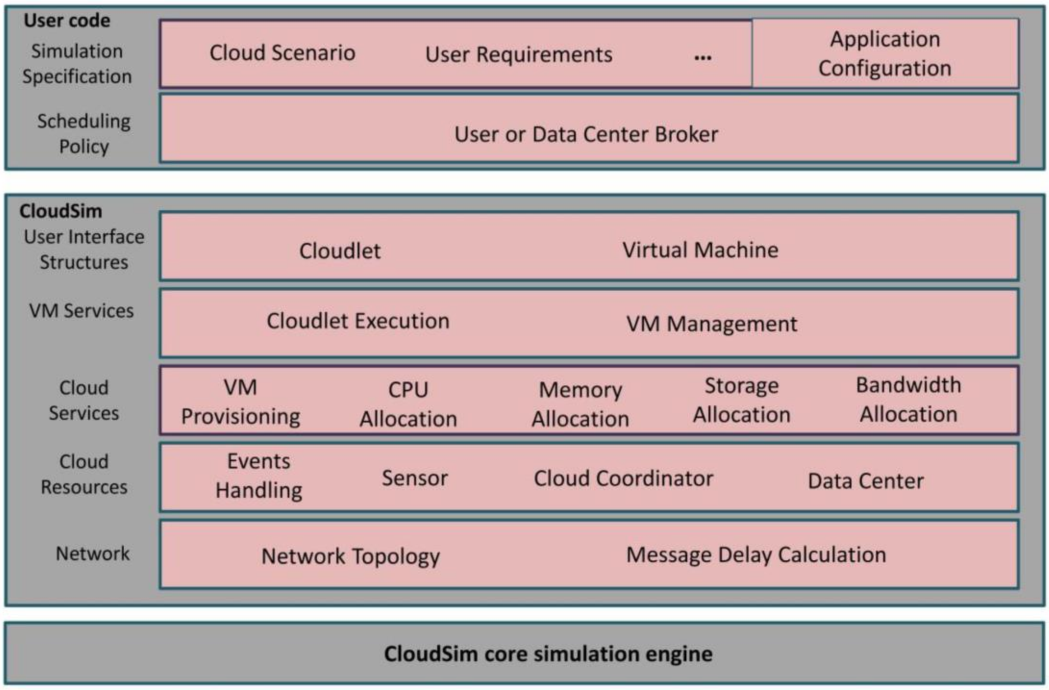

Fig. 5 Layered CloudSim architecture [2].

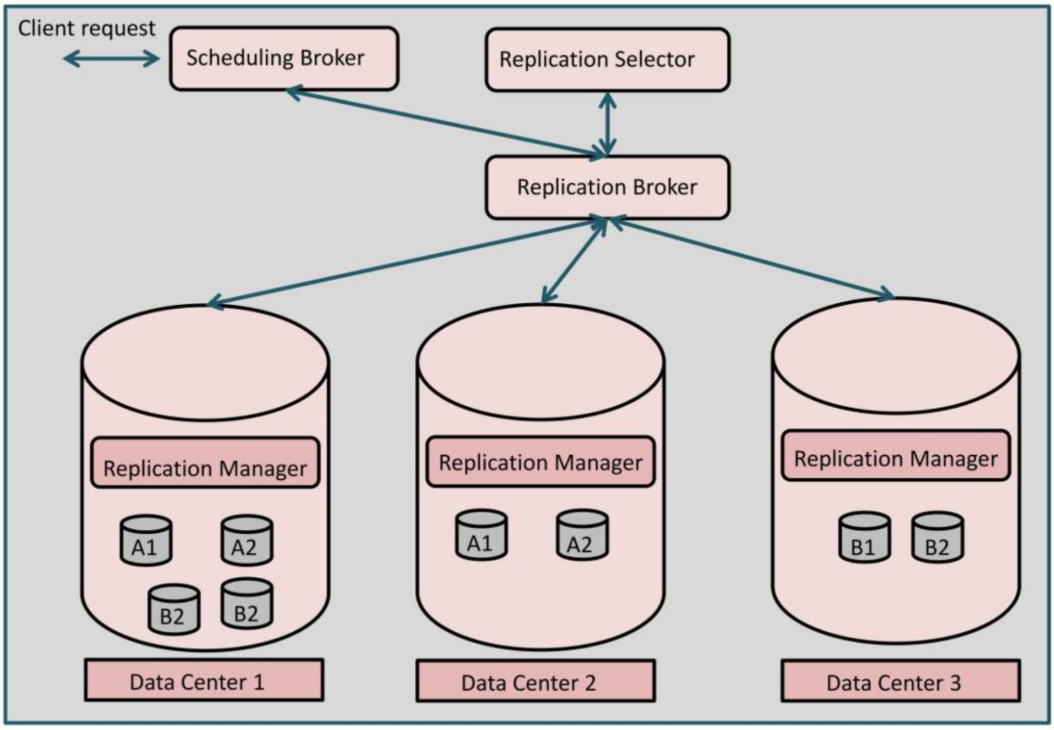

Fig. 6 The Cloud data server architecture [2].

$$ART = \frac{\sum_{j=1}^{m}\sum_{k=1}^{m} j\left(ts_{jk}\left(rt\right) - ts_{jk}\left(st\right)\right)}{\sum_{j=1}^{m} m_j} \qquad (16)$$

In which $ts_{jk}(st)$ and $ts_{jk}(rt)$ are the submission and the return times of the results for task k and user j, respectively. m_j is the number of the tasks of user j.

Fig. 7 compares the response time of the 7 dynamic replication strategies by various numbers of tasks. Accordingly by increasing the task number, the response time is increased. This trend is more serious for jobs higher than 1000 jobs. Typically, for task number equal to 1000; the response time of ARS and

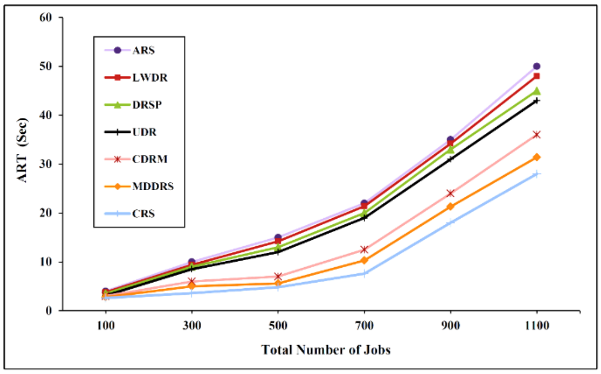

Fig. 7 Average Response Time for different replication algorithms.

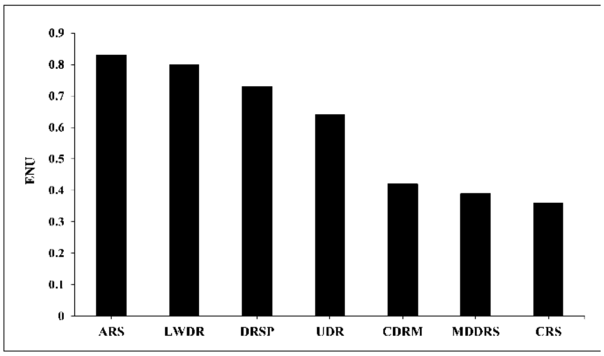

Fig. 8 Effective network usage for different replication algorithms.

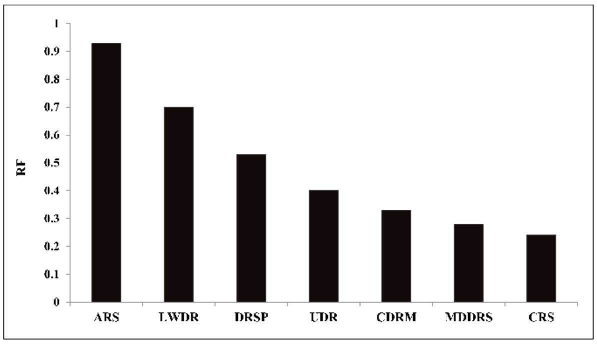

Fig. 9 Replication frequency for different replication algorithms.

CRS are 49 and 24, respectively. Accordingly, CRS is able to delete non-performance crucial replicas to enhance the free space for saving of new replica. The response time of DRSP is higher than 26% respect to MDDRS in 1000 jobs number.

7.2 Effective Network Usage (ENU)

ENU is used to estimate the efficiency of network resource usage. Effective Network Usage (E_{enu}) is given from [39]:

$$E_{enu} = \frac{N_{rfa} + N_{fa}}{N_{lfa}} \tag{17}$$

In which N_{rfa} is the number of access times that computing element reads a file from a remote site, N_{fa} is the total number of file replication operation, and N_{lfa} is the number of times that computing element reads a file locally. E_{enu} changes in the range of 0-1. By decreasing of E_{enu} to 0; the efficiently of bandwidth application is enhanced.

In spite of time and network bandwidth consumption, simple replication strategy is more effective than no replication. Therefore, reasonable methodology must be improved to balance in a way that any replication caused to reduce the traffic of network in the future. Lower value of ENU, i.e., the ratio of transferred files to requested file, indicated that the proposed strategy replicates candidate file to the better location.

By consideration of ENU (Fig. 8), CRS and ARS have the lowest and highest ENU values, respectively. While, the ENU of LWDR and MDDRS are 0.8 and 0.39. With the CRS, sites will contain necessary files at the time of need, hence the total number of replications will decrease and total number of local accesses increase. Accordingly, CRS effectively minimizes the bandwidth consumption as well as network traffic.

7.3 Replication Frequency (RF)

By decreasing of the replication frequency, i.e., the ratio of how many replication trigger per data access, the ability of replication algorithm to sort data file in the appropriate sites is improved. The results of the replication frequency are presented in Fig. 9. The replication creation is higher than 0.92 for ARS algorithm, which shows that at least 0.92 replicas are created for a data access. The replication frequency of LWDR method is too high which renders it not feasible in the real environment. Fig. 9 shows that MDDRS strategy has reasonable replica frequency, it is due to fact that it creates replicas on the basis of popularity degree. The replication frequency of CRS is less than 0.24, i.e., for successive 100 data access; 24 replicas must be created. The ability of CRS is to save valuable replicas during the change of environment. These characteristics can increase the availability beside of reducing unnecessary replications. Since CRS transfers a small number of files therefore it does not affect network communication cost considerably.

7.4 Degree of Imbalance (DI)

For the third experiment, we compared the average degree of imbalance, which measures the imbalance among virtual machines. Degree of imbalance is a metric that has close relation to the system load balancing. Degree of imbalance indicates whether the jobs are distributed monotonously among virtual machines or not. Considering Fig. 10, it can be seen that CDRM method performs better than UDR and DRSP with respect to the average degree of imbalance. This is because it creates more replicas when the value of

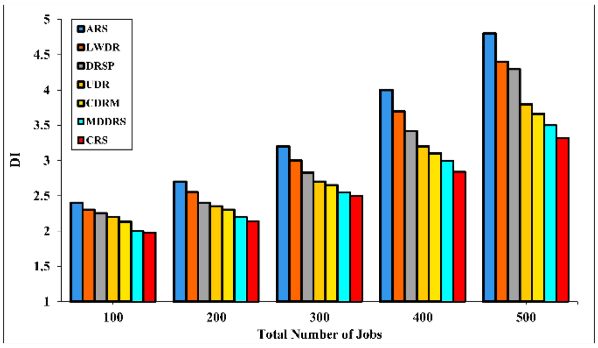

Fig. 10 Degree of imbalance for different replication algorithms.

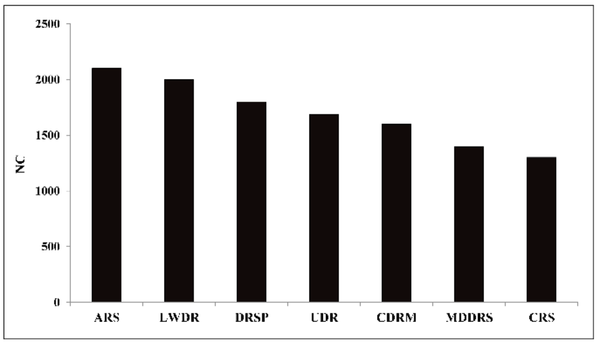

Fig. 11 Number of Communications for different replication algorithms.

availability is lower than the proposed threshold. Meanwhile, we could also notice that degree of imbalance of CRS algorithm decrease with 9% in comparison to the MDDRS strategy. The main reason is that replicas are placed in different sites based on the frequency of requests, access time, and centrality.

7.5 Number of Communications (NC)

It is crucial to reduce the total number of communications for decreasing the data access latency and preventing the bandwidth congestion. Comparing the curves shown in Fig. 11, it is observed that CRS outperforms by 7% over MDDRS and by 15% over CDRM. This is because, CRS stores replicas in the best site (i.e., best site that has the great number of access and most central) based on the temporal and geographical locality concepts. Consequently, it can decrease total communications between data centers.

8 Conclusion

Cloud data management is increasingly attracting researchers' attention. This study considers data replication as a powerful technique for decreasing of user waiting time and increasing the data availability. Accordingly, the proposed Combined Replication Strategy (CRS) efficiently distributes workload across cloud nodes and improves the response time. Due to restricted storage space, a replica replacement method is necessary to make the dynamic replica management efficient. CRS replaces replicas based on the number of access file, the last time the replica was requested, number of access and size of replica as criteria. Simulation results using CloudSim platform confirms that CRS respect to some recent reported algorithms, i.e., ARS, LWDR, DRSP, CDRM, UDR and MDDRS), reduces the number of data replication as well as data transfer time, and bandwidth usage. For example, CRS provides a 38% decrease in average response time over MRU for 1000 jobs. The reason behind that ARS strategy did not take into account the centrality and probability in replica placement, so, the selected data center is not always the best replica providers. Furthermore, compared with UDR and CDRM, the replication frequency of CRS can reduce up to 35% and 24% respectively. Since CRS uses some affecting factors, i.e., frequency of requests of replica, the last time of requesting time, failure probability, centrality, and storage usage as threshold for selection of best site. Especially, by increasing of task number, the performance of CRS increased with higher rate than the others. In the future, In the future, we want to investigate the HRS reliability and focus on the temporal fault tolerance. Moreover, the replication strategy will be implemented and checked on a real cloud-computing scheme.

References

[1] X. Yang, D. Wallom, S. Waddington, J. Wang, A. Shaon, B. Matthews, M. Wilson, Y. Guo, L. Guo, J. Blower, A. V. Vasilakos, K. Liu, and P. Kershaw, "Cloud computing in e-science: research challenges and opportunities," *The Journal of Supercomputing*, Vol. 70, No. 1, pp. 408–464, 2014.

[2] T. Hamrouni, S. Slimani, and F. Ben Charrada, "A survey of dynamic replication and replica selection strategies based on data mining techniques in data grids," *Engineering Applications of Artificial Intelligence*, Vol. 48, pp. 140–158, 2016.

[3] N. Mansouri, "Adaptive data replication strategy in cloud computing for performance improvement," *Frontiers of Computer Science*, Vol. 10, No. 5, pp. 925–935, 2016.

[4] M. Mahdavi Jafari and G. R. Khayati, "Prediction of hydroxyapatite crystallite size prepared by sol–gel route: gene expression programming approach," *Journal of Sol-Gel Science and Technology*, Vol. 86, No. 1, pp. 112–125, 2018.

[5] N. Mansouri and G. H. Dastghaibyfard, "Enhanced dynamic hierarchical replication and weighted scheduling strategy in data grid," *Journal of Parallel and Distributed Computing, Journal of Parallel and Distributed Computing*, Vol. 73, No. 4, pp. 534–543, 2013.

[6] N. Mansouri, "A threshold-based dynamic data replication and parallel job scheduling strategy to enhance data grid," *Cluster Computing*, Vol. 17, No. 3, pp. 957–977, 2014.

[7] W. Dai, I. Ibrahim, and M. Bassiouni, "An improved replica placement policy for Hadoop distributed file system running on cloud platforms," in *IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 270–275, 2017.

[8] H. Luo, J. Liu, and X. Liu, "A two-stage service replica strategy for business process efficiency optimization in community cloud," *Chinese Journal of Electronics*, Vol. 26, No. 1, pp. 80–87, 2017.

[9] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: amazon's highly available key-value store" in *21st ACM Symposium on Operating Systems Principles*, pp. 205–220, 2007.

[10] Y. Qu and N. Xiong, "RFH: a resilient, fault-tolerant and high-efficient replication algorithm for distributed cloud storage," in *41st International Conference on Parallel Processing (ICPP)*, pp. 520–529, 2012.

[11] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience (SPE)*, Vol. 41, No. 1, pp. 23–50, 2011.

[12] L. Peng, *The definition of cloud computing and characteristics*, http:// www.chinacloud.cn/.2009-2-25.

[13] IBM Center for High Performance On Demand Solutions, *IBM cloud computing white paper*, http://www.ibm.com/developerworks/websphere/zones/hipods.

[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwin-ski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, Vol. 53, No. 4, pp. 50–58, 2010.

[15] P. Mell and T. Grance, "The NIST definition of cloud computing," *Communications of the ACM*, Vol. 53, 2010.

[16] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, pp. 931–945, 2011.

[17] M. Tao, K. Ota, and M. Dong, "Dependency-aware dependable scheduling workflow applications with active replica placement in the cloud," *IEEE Transactions on Cloud Computing*, Vol. 99, 2017.

[18] H. Wang, "Privacy-preserving data sharing in cloud computing," *Journal of Computer Science and Technology*, Vol. 25, No. 3, pp. 401–414, 2010.

[19] A. Gupta and R. Garg, "Load balancing based task scheduling with ACO in cloud computing," in *International Conference on Computer and Applications (ICCA)*, pp. 174–179, 2017.

[20] B. M. Xu, C. Y. Zhao, E. Z. Hu, and B. Hu, "Job scheduling algorithm based on Berger model in cloud environment," *Advances in Engineering Software*, Vol. 42, pp. 419–425, 2011.

[21] N. Ahmad, A. A. C. Fauzi, R. M. Sidek, N. M. Zin, and A. H. Beg, "Lowest data replication storage of binary vote assignment data grid," in *the 2nd International Conference Networked Digital Technologies*, pp. 466–473, 2010.

[22] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "CDRM: A cost effective dynamic replication management scheme for cloud storage cluster," in *IEEE International Conference on Cluster Computing*, pp. 188–196, 2010.

[23] M. Hussein and M. H. Mousa, "A light-weight data replication for cloud data centers environment," *International Journal of Engineering and Innovative Technology (IJEIT)*, Vol. 1, pp. 169–175, 2012.

[24] A. Rajalakshmi, D. Vijayakumar, and K. G. Srinivasagan, "An improved dynamic data replica selection and placement in cloud," in *International Conference on Recent Trends in Information Technology*, pp. 1–6, 2014.

[25] M. Hussein and M. H. Mousa, "A light-weight data replication for cloud data centers environment," *International Journal of Engineering and Innovative Technology (IJEIT)*, Vol. 1, pp. 169–175, 2012.

[26] D. W. Sun, G. R. Chang, Sh. Gao, L. Jin, and X. W. Wang, "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.

[27] L. Chen and D. B. Hoang, "Adaptive data replicas management based on active data-centric framework in cloud environment," in *IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing*, pp. 101–108, 2013.

[28] H. Zhan, B. Li, Zh. Liu, and W. Guo, "Data replication placement strategy based on bidding mode for cloud storage cluster," in: *Web Information System and Application Conference*, pp. 207–212, 2014.

[29] N. Mansouri, "A utility-based data replication algorithm in large scale data grids," *Journal of Advances in Computer Research*, pp. 79–93, 2015.

[30] S. Kirubakaran, S. Valarmathy, C. Kamalanathan, "Data replication using modified D2RS in cloud computing for performance improvement," *Journal of Theoretical and Applied Information Technology*, Vol. 58, No. 2, pp. 460–470, 2013.

[31] S. Ghemawat, H. Gobioff and S. Leung, "The Google File System," in *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP)*, 2003.

[32] K. Shvachko, K. Hairong, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *Proc. the 26th Symposium on Mass Storage Systems and Technologies*, pp. 1–10, 2010.

[33] T. Kwan, R. Mcgrath, and D. Reed, "NCSAs World Wide Web server design and performance," *Computer*, Vol. 28, pp. 67–74, 1995.

[34] X. T. SEA, "A striping-based energy-aware strategy for data placement in RAID-structured storage systems," *IEEE Transactions on Computers*, Vol. 57, pp. 748–761, 2008.

[35] A. Saleh R. Javidan, and M. T. Fatehikhaje, "A four-phase data replication algorithm for data grid," *Journal of Advanced Computer Science & Technology*, Vol. 4, 2015.

[36] M. Newman, *Networks: An introduction*, Oxford University Press, 2009.

[37] M. Lei, S. V. Vrbsky, and X. Hong, "An on-line replication strategy to increase availability in data grid," *Future Generation Computer Systems*, Vol. 28, pp. 85–98, 2008.

[38] F. Howell and R. Mcnab, "A discrete event simulation library for java," in *Proceedings of the First International Conference on Web-based Modeling and Simulation*, 1998.

[39] D. G. Cameron, R. Carvajal-Schiaffino, A. Paul Millar, C. Nicholson, K. Stockinger, and F. Zini, "UK Grid Simulation with OptorSim," *UK e-Science All Hands Meeting*, 2003.

N. Mansouri is currently a faculty of Computer Science at Shahid Bahonar University of Kerman. She received her Ph.D. (Honor Student) in Computer Science from Shahid Bahonar University of Kerman and M.Sc. in Software Engineering at Department of Computer Science & Engineering, College of Electrical & Computer Engineering, Shiraz University, Iran. She received her B.Sc. (Honor Student) in Computer Science from Shahid Bahonar University of Kerman, in 2009. She has published more than 40 scientific papers in the field of high-performance, parallel processing, grid and cloud computing, and contributed to more than 20 research and development programs.

M. M. Javidi is currently an Associate Professor of Computer Science in Department of Computer Science, Shahid Bahonar University of Kerman. His research interests include distributed systems, and cloud computing. He has published more than 50 articles in journals and conferences.