Journal of Network and Computer Applications 64 (2016) 229–238

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions

Bahareh Alami Milani [1], Nima Jafari Navimipour *

Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

ARTICLE INFO

Article history:
Received 28 June 2015
Received in revised form
9 January 2016
Accepted 11 February 2016
Available online 23 February 2016

Keywords:
Cloud computing
Replication
Big data
Static
Dynamic

ABSTRACT

Nowadays, in various scientific domains, large data sets are becoming an important part of shared resources. Such huge mass of data is usually stored in cloud data centers. Therefore, data replication which is generally used to manage large volumes of data in a distributed manner speeds up data access, reduces access latency and increases data availability. However, despite the importance of the data replication techniques and mechanisms in cloud environments, there has not been a comprehensive study about reviewing and analyzing its important techniques systematically. Therefore, in this paper, the comprehensive and detailed study and survey of the state of art techniques and mechanisms in this field are provided. Also, we discuss the data replication mechanisms in the cloud systems and categorize them into two main groups including static and dynamic mechanisms. Static mechanisms of data replication determine the location of replication nodes during the design phase while dynamic ones select replication nodes at the run time. Furthermore, the taxonomy and comparison of the reviewed mechanisms are presented and their main features are highlighted. Finally, the related open issues and some hints to solve the challenges are mapped out. The review indicates that some dynamic approaches allow their associated replication strategies to be adjusted at run time according to changes in user behavior and network topology. Also, they are applicable for a service-oriented environment where the number and location of the users who intend to access data often have to be determined in a highly dynamic fashion.

Contents

* Corresponding author. Tel.: +98 9144021694.
E-mail address: jafari@iaut.ac.ir (N. Jafari Navimipour).
[1] Tel.: +98 9144126036.

http://dx.doi.org/10.1016/j.jnca.2016.02.005
1084-8045/

1. Introduction

Cloud computing is a network-based infrastructure where information technology (IT) and computing resources such as operating systems, storage, networks, hardware, databases, and even entire software applications are delivered to users as on-demand services (Buyya et al., 2008). Cloud computing does not consider a lot of new technologies, however, it saves the cost and increases the scalability to manage IT services (Buyya and Ranjan, 2010). The provided in cloud computing are grouped into 4 categories, including Software as a Service (SaaS) (Almorsy et al., 2014; Buxmann et al., 2008; Choudhary, 2007; Lin et al., 2009; Zeng and Veeravalli, 2014), Infrastructures as a Service (IaaS) (Bhardwaj et al., 2010; Iosup et al., 2014; Khajeh-Hosseini et al., 2010; Lin et al., 2009; Nathani et al., 2012; Wang et al., 2013; Zeng and Veeravalli, 2014), Platforms as a Service (PaaS) (Dinesha and Agrawal, 2012; Eludiora et al., 2011; Lin et al., 2009; Mell and Grance, 2009; Miller and Lei, 2009; Sellami et al., 2013; Zeginis et al., 2013; Zeng and Veeravalli, 2014) and Expert as a Service (EaaS) (Ashouraie et al., 2015; Nima Jafari Navimipour and Milani, 2015; Nima Jafari Navimipour, 2015; Nima Jafari Navimipour et al., 2015a, 2015b; Oussalah et al., 2014).

On the other hand, currently, in different scientific disciplines, an enormous amount of data is an important and vital part of shared resources. The mass of data is measured in terabytes and sometime in petabytes in many fields. Such enormous mass of data is typically kept in the cloud data centers (Long et al., 2014). So, data replication is generally used to manage a great deal of data (Wolfson et al., 1997) by creating identical copies of data (files, databases, etc.) in geographically distributed sites, which are called replicas (Lamehamedi and Szymanski, 2007; Meroufel and Belalem, 2013). The advantage of data replication is speeding up data access, reducing access latency and increasing data availability (Berl et al., 2010; Long et al., 2013). A general method is using multiple replicas which are distributed in geographically-dispersed clouds to increase the response time to users. It is important to guarantee replica's availability and data integrity features; i.e., the same as the original data without any interfering and corruption. Remote data ownership checking is an effective method to prove the replica's availability and integrity (He et al., 2012). Replication is one of the most broadly studied phenomena in the distributed environments (Goel and Buyya, 2006) in which multiple copies of some data are stored at multiple sites where overheads of creating, maintaining and updating the replicas are important and challenging issues (Dayyani and Khayyambashi, 2013; Goel and Buyya, 2006).

Since data replication is coming to play an increasingly important role in the cloud, the purpose of this paper is to survey the existing techniques and to outline the types of significant challenges and issues that can be addressed in the cloud replication domain. To the best of our knowledge, this survey paper is a first attempt to comprehensively and systematically examine the data replication problem with a specific focus on the cloud. The contributions of this paper are as follows:

- Providing the basic concepts and terminologies which are used in the field of data replication.
- Discussing the data replication mechanisms in the cloud systems and categorizing them into two main groups including static and dynamic mechanisms.
- Presenting the taxonomy and comparison of the reviewed mechanisms and highlighting their features.
- Mapping out the related open issues and some hints to solve the existing problems.

The rest of this paper is structured as follows. Section 2 discusses the data replication approaches in a cloud environment and classifies them. Section 3 presents the taxonomy and comparison of the reviewed mechanisms. Section 4 maps out some open issues. At last, Section 5 comes up with the conclusion of this paper.

2. Data replication mechanisms

Replication has been an area of interest for many years in World Wide Web (Qiu et al., 2001), peer-to-peer networks (Aazami et al., 2004; Nima Jafari Navimipour and Milani, 2014), ad-hoc and sensor networking (Intanagonwiwat et al., 2000; Tang et al., 2008), and mesh networks (Jin and Wang, 2005). Replication is a strategy that creates multiple copies of some data and stored them at multiple sites (Goel and Buyya, 2006). Data replication is a technique which is used in the cloud to decrease the user waiting time, to increase data availability and to minimize cloud system bandwidth consumption utilizing different replicas of the same service (Ahmad et al., 2010). More recently, the emergence of large-scale distributed systems such as Grid (Dabrowski, 2009; Nima Jafari Navimipour et al., 2014; Navin et al., 2014; Souri and Navimipour, 2014) and cloud (Ashouraie et al., 2015; Bonvin et al., 2009; Jafari Navimipour et al., 2014; Nima Jafari Navimipour and Milani, 2015; Talia et al., 2016) has made data replication becoming a research hot spot once again. In data clouds, enormous scientific data and complex scientific applications require different replication algorithms, which have attracted more attention recently. Data replication techniques can be classified into two main groups including static and dynamic replication mechanisms that are shown in Fig. 1. The number of replicas and the host node is predetermined and well-defined in the static replication strategies (Ghemawat et al., 2003; Rahman et al., 2006; Shvachko et al., 2010). Whereas, dynamic strategies automatically create and remove replicas based on the changes in user access pattern, storage capacity and bandwidth (Chang and Chang, 2008; Doğan, 2009; Lei et al., 2008; Li et al., 2011; Wei et al., 2010). It makes intelligent choices about the location of data depending upon the information of the current situation. But, it has some drawbacks such as difficulty to collect runtime information of all the data nodes in a complex cloud infrastructure and hard to maintain consistency of data file (Long et al., 2014). Static and dynamic replication algorithms can be further classified into groups as distributed (Doğan, 2009; Ghemawat et al., 2003; Shvachko et al., 2010; Wei et al., 2010) and centralized algorithms (Chang and Chang, 2008; Lei et al., 2008; Rahman et al., 2006; Sun et al., 2012).

2.1. Static mechanisms

In this section, the static mechanisms of data replication and their basic properties are described. Then, eight most popular static mechanisms of data replication are discussed. Finally, these mechanisms are compared and summarized in Section 2.1.3.

2.1.1. Overview of the static strategies

Static replication strategies follow deterministic policies, therefore, the number of replicas and the host node is well-defined and predetermined (Long et al., 2014). Also, these strategies are simple to implement but it is not often used because it does not adapt according to the environment (Gill and Singh, 2015). In the next sub-section, some applicable and popular static data replication mechanisms in the cloud environments are reviewed and discussed.

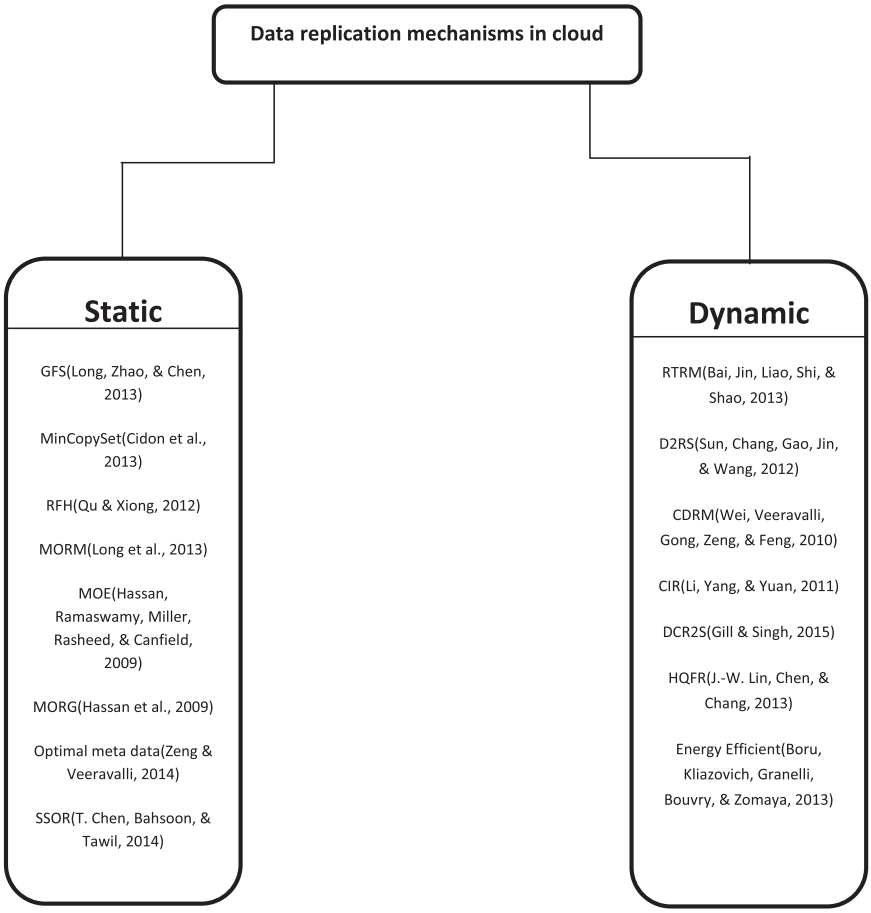

Fig. 1. Data replication mechanisms in the cloud environment.

2.1.2. Popular static mechanisms

Ghemawat et al. (2003) have proposed a Google File System (GFS) method to grab data replication where there are two vital questions in data replication in cloud storage clusters that must be solved: (1) How many suitable replicas of each data should be created in the cloud to match a reasonable system requirement; (2) Where should these replicas be situated to meet the system task fast execution rate and load balancing requirements. These two correlated issues are referred to as the replica management problem (Ghemawat et al., 2003). Static method for replication provides fast response, high availability, and high efficiency. However, the data replication does not come for free, and it uses several resources like storage and energy. Additionally, as the number of replicas increased, the energy consumption increases as well. For decreasing the energy cost, the number of replicas should be as small as possible. GFS considers certain factors when making choices on data chunk replications: insertion the new replicas on chunk servers with below-average disk space utilization, limiting the number of recent creations on each chunk server, and spreading replicas of a chunk across racks. The limitation of this algorithm is that a fixed replica number is used for all files which may not be the best solution for data (Long et al., 2014).

Cidon et al. (2013) have proposed a MinCopysets method which is a simple general purpose scalable replication scheme. It derandomized the data replication in order to achieve better data durability properties. MinCopysets has decoupled the mechanisms used for data distribution and durability. It allows system designers to use randomized node selection for data distribution to reap the benefits of parallelization and load balancing. MinCopyset does not present any noteworthy overhead on normal storage operations, and can support any data locality or network topology requirements of the underlying storage system (Cidon et al., 2013). The servers are divided statically into replication groups (each server belongs to a single group). When a chunk has to be replicated, a primary node is selected randomly for its first replica, and the other replicas are stored deterministically on secondary nodes, which are the other nodes of the primary node's replication group MinCopysets provides significant improvements in data durability and increases the network latency and disk bandwidth of write operations (Cidon et al., 2013).

As another static mechanism, Qu and Xiong (2012) have proposed Resilient, Fault-tolerant, and High-efficient (RFH) method to achieve high availability while maintaining low replication cost. This algorithm can adapt the replica number according to varying traffic. If a partition becomes hot, more replicas will be replicated to meet the need of service, or replicas will migrate to reach higher utilization. Otherwise, unwanted replicas will commit suicide to save resources. The advantage is relatively lower replication cost and consequently lower replication failure possibility. However, the lookup path length and response time cannot be significantly reduced, especially when most of a lot of queries is from far away continents. Another method is request-oriented, which encourages replicating data on data centers near to the requesters with the highest query rate. This method reduced lookup path length dramatically and improved query efficiency. However, it cannot guarantee replica utilization rate since those other requesters will have a lower chance to access these replicas. It can cause a massive increase in traffic within a few minutes, and it can also pass into silence after peak time. This algorithm is a better way to address this challenge, which can achieve high replica utilization rate and

high query efficiency while maintaining high availability and reasonable path length at a low cost. Rather than owner-oriented or request-oriented, it replicates data on nodes with the most forwarding traffic (Qu and Xiong, 2012).

Long et al. (2013) have proposed a Multi-objective Optimized Replication Management (MORM) strategy for cloud storage cluster using the artificial immune algorithm. Five factors were considered including mean file unavailability, mean service time, load variance, energy consumption and mean access latency to capture trade-off between these factors and the relationship among replica number, replica layout and the performance (Long et al., 2014). MORM is an offline replication algorithm which makes capacity-aware random service placement. It creates an individual randomly which means that both layout of files and the replication factor are random (Long et al., 2014). The MORM needs the full knowledge of service time and access rates for all assigned files. It applies to the scene that access statistics is immutable, and, therefore, the replication scheme desires to be computed once only and can uninterruptedly work for a long period of time (Xie and Sun, 2009). But, when files arrive in storage uninterruptedly and dynamically, MORM is not practical since static file assignment algorithms are not applicable in such conditions. Particularly, when files are arriving in batches (such as computing and web applications), an additional algorithm to allocate the files based on the information about the coming batch of files and previously assigned files is required. MORM as a static replica management strategy is extended to dynamically replica management strategy, DMORM, which is proposed by Lee et al. (2000).

Also, Hassan et al. (2009) have proposed Multi-Objective Evolutionary (MOE) algorithm for data replication in large-scale cloud storage cluster using an evolutionary method to find the best replication strategy. Evolutionary computing is the branch of science that takes randomness as a mean of problem solving; it also considers solutions of the problem as populations. Evolutionary computing jumps in the search space in such a way that explores areas in which a potentially good solution can be found. Many of evolutionary computing techniques rely on operators such as crossover operator, mutation operator, and parent selection. MOE improves storage, latency, and reliability of the system and did not take total data center energy cost as the major optimization target (Long et al., 2014). It depends on taking the historical system information and feed it to an engine where does not try only to keep reliability, latency, and storage within limitations, but, in addition, tries to optimize latency, reliability, and storage, in order to find different trade-offs between these objectives using more than one objective (Hassan et al., 2009).

Zeng and Veeravalli (2014) have presented an optimal load balancing technique for large-scale cloud data centers that are composed of thousands of Raw Data Server (RDS) and hundreds of Meta Data Servers (MDS) connected by arbitrary networks. The purpose of the technique is to achieve the minimum Mean Response Time (MRT) of the metadata requests, which is one of the most significant performance indexes which shall be considered by data service providers. This technique models MDS as an M/M/1 system, that formulate objective function for the MRT of the metadata requests arriving at MDS. For each object, it has a master MDS, which takes responsibility of the metadata requests. The master MDS uses a heuristic way to find other MDS to store a replica of the metadata and construct the set of MDS for the object. The number of replicas of each object is based on the demand rate for the object, for example, some hot objects can have several replicas to meet the data requirements, while some cold object may have no replica at all. Near-optimal solutions can be achieved by this technique with hashing functions of 0, 1, 2, 3 replicas. It clearly offers an important benefit in minimizing the MRT, and at the same time load balancing of the MDS across the system with a minimum number of replicas. This technique handles metadata replication and a load balancing strategy to minimize MRT and demonstrate a trade-off relationship between makespan and the monetary cost (Zeng and Veeravalli, 2014).

Finally, Chen et al. (2014) have proposed Scalable Service-Oriented Replication (SSOR) middleware solution capable of satisfying consistency requirements in service-oriented and cloud-based applications. To address the important differences between Service Oriented Replication (SOR) and Data Oriented Replication (DOR), SSOR provides new formalisms to define services in SOR. SSOR can be used to ensure the consistency and to manage consistency requirements for collections of services within a region. To assure the different consistency models at runtime, defined new policies operate within regions. Also, to realize notions, formalisms, and policies two novel protocols are improved, Multi-fixed Sequencer Protocol (MSP) and Region-based Election Protocol (REP). MSP is responsible for warranting the satisfaction of the consistency in a region while REP aims at balancing the workload amongst sequences through electing new sequences upon the existence of failure and allocates the loads to multiple sequencers. SSOR increases flexible consistency which can lead to the improvement in the scalability of the managed cloud-based applications. The caused impact by the crash is significantly reduced with the distribution of services into different regions in SSOR. This implies that SSOR could also be useful for improving elastically in the cloud.

2.1.3. Summary of static mechanisms

In the static replication model, the replication strategy is predetermined and well-defined. Statically creating a maximum number of service replicas may guarantee the needed performance at a high operation cost, i.e., the cost is directly proportional to the number of active service replicas (Chen et al., 2005; M. Lin et al., 2013). The static replication strategy keeps the number of active service replicas at the maximum with a random policy (Bjorkqvist et al., 2011). The side-by-side comparison of the discussed static data replication mechanisms in a cloud environment is summarized in Table 1.

2.2. Dynamic mechanisms

In this section, the dynamic mechanisms for data replication and their basic properties and features are described. Then, seven most popular dynamic mechanisms of data replication are discussed. Finally, these mechanisms are compared and summarized in Section 2.2.3.

2.2.1. Overview of dynamic strategies

Dynamic strategies for data replication in cloud environments automatically create and delete the replicas according to changes in user access pattern, storage capacity and bandwidth (Chang and Chang, 2008; Doğan, 2009; Lei et al., 2008; Li et al., 2011; Wei et al., 2010). They make intelligent choices about the location of data depending upon the information of the current environment. But, it has some drawbacks such as difficulty to collect runtime information of all the data nodes in a complex cloud infrastructure and maintaining the data file consistency (Long et al., 2014). Dynamic data replication strategies include some phases: analyzing and modeling the relationship between the number of replicas and system availability; recognizing the popular data and triggering a replication process when the data passes a dynamic threshold; evaluating a suitable number of copies to meet a reasonable system byte effective rate requirement and insertion replicas among data nodes in a balanced way; and designing the dynamic data replication algorithm in a cloud. In a dynamic replica management strategy, replica creation component decides which file has the

Table 1
Side-by-side comparison of the static data replication mechanisms in cloud environment.

Mechanisms	Main idea	Advantage	Disadvantage
GFS (Long et al., 2013)	It considers three factors: insertion the new replicas with below-average disk space utilization, limiting the number of recent creations on each server, and spreading replicas of an across racks.	• High availability • Low response time • High reliability • Medium load balancing	• High energy • High replication cost • High storage cost • High bandwidth consumption
MinCopysets (Cidon et al., 2013)	It improves data durability using the benefits of randomized load balancing.	• High data durability • High availability • Low storage cost	• High energy consumption • High bandwidth consumption • High response time • Low reliability
RFH (Qu and Xiong, 2012)	It can adapt the replica number according to changing traffic.	• High reliability • High fault-tolerant • High availability • Low bandwidth consumption • Low replication cost	• High energy consumption • High response time
MORM (Long et al., 2014)	It obtains the near optimal solutions by balancing the trade-off among the mean service time, mean file unavailability, load variance mean access latency, and energy consumption.	• Low storage cost • High availability • Low response time • High reliability • Low energy consumption	• High bandwidth consumption
MOE (Long et al., 2014)	It employs an evolutionary way to find the optimal replication strategy.	• High scalability • High performance • Low access latency • Low storage cost	• High energy consumption • High replication cost • Low availability • High execution time • Low reliability
MORG (Hassan et al., 2009)	It is the greedy algorithm and has the different start point to explore replication node.	• High scalability • High performance • Low access latency • Low execution time	• High energy consumption • High replication cost • High bandwidth consumption
Optimal metadata replications (Zeng and Veeravalli, 2014)	It presented an optimal load balancing strategy for large-scale cloud data centers that are composed of hundreds of MDS and thousands of RDS connected by arbitrary networks.	• Low response time • High load balancing • Low delay of path • High scalability	• High replication cost
SSOR (Chen et al., 2014)	It is an SSOR middleware solution that is capable of satisfying consistency requirements in service-oriented and cloud-based applications.	• High scalability • High availability • High load balancing	• High latency • High replication cost

popular data and when is the right time to create a new replica of the popular data. Replica creation method first finds the best time to create a new replica, an access recorder is assigned to each data node, which is used to store the number of simultaneous users accesses to each file, including file name, a number of concurrent access, file size, and so on (Bai et al., 2013). In the next sub-section, some popular and applicable dynamic replication mechanisms in the cloud environments are discussed and analyzed.

2.2.2. *Popular dynamic mechanisms*

In 2013, Bai et al. (2013) have proposed a response time-based replica management strategy referred to as Response Time-Based Replica Management (RTRM) to create a replica for automatically enhancing the number of replicas based on the average response time. RTRM will predict the bandwidth among the replica servers upon receiving the new request, makes the replica selection accordingly, and replica placement mechanism combining the number of replicas and the network transfer time (Bai et al., 2013). RTRM strategy consists of three levels, replica creation, replica selection, and replica placement. RTRM sets a threshold for response time, if the response time is longer than the threshold, it creates a new replica and increases the number of replicas. RTRM will predict the bandwidth among the replica servers upon receiving the new request, and makes the replica selection accordingly. The simulation results showed that RTRM strategy improves the replica management strategies in terms of network utilization and service response time (Bai et al., 2013).

Sun et al. (2012) have proposed Dynamic Data Replication Strategy (D2RS) strategy in three important phases: 1) which and when data file should be replicated in the cloud system to meet users' requirements such as waiting time deduction and data access acceleration; 2) how many suitable new replicas should be made in the cloud system to meet a given availability requirement; 3) where the new replicas should be placed to meet the system function successful execution rate and bandwidth consumption requirements (Sun et al., 2012). A popular data file is determined by the analysis of the access information to the data from users. When the popularity of a data file passes a dynamic threshold, the replication process will be triggered. The number of replicas depends on the reasonable growth of file availability. Also, the replica placement is determined by the access information of directly connected data centers and is accomplished in a balanced way. It reduced the user waiting time, speeded up the data access and increased the data availability by providing the users different replicas of the same service (Sun et al., 2012).

Also, Wei et al. (2010) have proposed a new model to grab the relationship between replica number and availability named Cost-effective Dynamic Replication Management (CDRM). CDRM computes and keeps minimal replica number for a given availability condition. Data replication has been broadly used as a mean of increasing the data availability of large-scale cloud storage environments where failures are normal. CDRM is a cost-effective dynamic replication management system aiming at providing cost effective availability, and improving the load balancing and performance of cloud storage. By regulating replica number and location according to workload changing and node capacity, CDRM can dynamically reallocate workloads among data nodes in the heterogeneous cloud. CDRM fulfills availability requirement at low

cost by dynamically maintaining minimum replicas among the system. A second main quality of this study is that it dynamically places replicas to distribute workload across cluster according to data node capacity and activity intensities. The data replica placement is based on the capacity and blocking probability of data nodes. CDRM improves access latency, load balancing, and keeps the whole storage system stable (Wei et al., 2010).

Li et al. (2011) have presented a cost-effective dynamic data replication strategy named Cost-effective Incremental Replication (CIR). CIR is a data reliability strategy for cloud-based applications in which the major focus is for cost-effectively managing the data reliability problem in a data center. In CIR, an incremental replication method is used for computing the time point of replica creation which shows the storage duration that the reliability requirement can be met. By predicting when an additional replica is needed to ensure the reliability requirement, CIR dynamically maximizes the number of replicas. In this way, the number of replicas can be minimized, so that the cost-effective dynamic data replication management goal can be reached. Utilizing a minimum number of replicas while meeting the data reliability requirement is the main idea of CIR strategy (Li et al., 2011). The result of the evaluation indicated that the CIR strategy can substantially reduce the number of replicas in a data center while meeting the reliability requirement, thus, the storage cost of the all storage system can be significantly reduced.

Gill and Singh (2015) have presented an algorithm named Dynamic Cost-aware Re-replication and Re-balancing Strategy (DCR2S) with the concept of knapsack problem to optimize the cost of replication. First, the system is designed to understand the relation between a number of replicas, cost of replication, and availability. Then, an algorithm determines which file needs to be replicated and when to replicate it. It also determines the suitable number of replicas and then places theses replicas in such a way that the cost should not increase more than the budget along with high system byte effective rate and bandwidth consumption. Furthermore, knapsack algorithm is used for improving the cost of replication. DCR2S has 3 phases: (1) determining which and when to replicate a data file to find the appropriate data file for replication and also decide when it should be replicated using the concept of temporal locality. According to this concept, recently retrieved data file has more probability of being accessed again in the future. Access history of each data file is analyzed to determine its respective acceptance. To determine the appropriate number of new replicas for a data file, replication operation is provoked as soon as its popularity crosses a dynamic threshold. (2) Determining additional required replicas to meet the available requirement. (3) This phase is related to the placement of new replicas. Experimental results proved that DCR2S can optimize the cost of replication and also achieve high system byte effective rate and is effective in heterogeneous cloud system architecture (Gill and Singh, 2015).

J.-W. Lin et al. (2013) have presented a greedy algorithm called High-QoS First-Replication (HQFR) algorithm which is considered a QoS-Aware Data Replication (QADR) problem for data-intensive applications in cloud systems. The main concern of QADR is to efficiently define the QoS requirements of applications in the data replication and to minimize the data replication cost. Minimizing the data replication cost reduces the data replication completion time which can significantly reduce the probability that the data corruption occurs prior to completion of data replication. The replicas of some applications may be stored in lower-performance nodes due to the inadequate replication space of a storage node. As a result, some data replicas cannot meet the QoS requirements of their corresponding applications. These data replicas are called the QoS-violated data replicas which are expected to be as small as possible. In HQFR, if the application has a higher QoS requirement, it will take precedence over other applications to operate data replication. However, the HQFR algorithm cannot achieve the above minimum objective. This is the scalable mechanism, but it does not have polynomial time complexity (J.-W. Lin et al., 2013).

Finally, Boru et al. (2013) have presented an energy-efficient data replication method in cloud data centers. In this replication approach, every data object is permanently stored at the Central Data Base (CentralDB) and depending on the access pattern, it is replicated in data center DB and Rack DBs. Any failures in data center DBs can be recovered from central DB and vice versa. Moreover, this approach implements a dynamic replication to improve both availability and the QoS of cloud applications which only maintain an optimal number of replicas. This approach decreases the energy and bandwidth consumption of the system. In addition, this is the promoted quality of QoS achieved as a result of the reduced communication delays. Also, this data replication technique improves communication delay and network bandwidth between geographically dispersed data centers as well as inside of each data center (Boru et al., 2013).

2.2.3. Summary of dynamic mechanisms

Dynamic strategies automatically make and omit replicas according to changes in user access pattern, storage capacity, and bandwidth. The effectiveness and efficiency of the system are determined by these methods in the cloud environments. It makes intelligent decisions about the situation of data depending upon the information of the current environment. Most dynamic replica management strategies create a new replica of the popular data based on the user access frequency, thus, the replica creation always happens at the end of each time interval (Bai et al., 2013). The side-by-side comparison of the discussed dynamic data replication techniques in a cloud environment is summarized in Table 3.

3. Results and comparison

Replication approaches in the cloud environments can be classified into two categories: static and dynamic approaches. Static approaches determine the locations of replica nodes during the design phase while dynamic ones determine the locations of replicas nodes at a run time. Some dynamic approaches even allow their associated replication strategies to be adjusted at run time according to changes in user behavior and network topology. Dynamic replication strategy is mainly more suitable for a service-oriented environment where the number and location of the users who is going to access data often have to be determined in a highly dynamic way (Boru et al., 2013).

As obtained from our review and analyze in the previous sections, the number of replicas is constant in static mechanisms and it does not adapt according to the environment. Therefore, these mechanisms are suitable for the environment that predetermines user's request. The reviewed mechanisms employed different strategies to improve the replication process efficiency. For example, GFS considers three factors when making decisions on data chunk replications: insertion new replicas on chunk servers with below-average disk space utilization, limiting the number of recent creations on each chunk server, and spreading replicas of a chunk across racks. It improves the availability, reduces the response time and enhances the load balancing. But, a static replica number is used for all files which may not be the best answer in cloud environments. On the other hand, MinCopysets is a general-purpose, simple and scalable replication technique to increase data durability while retaining the benefits of randomized load balancing. It uses randomized node selection for data distribution to reap the benefits of parallelization and load balancing.

Table 2
Side-by-side comparison of the dynamic data replication techniques in cloud environment.

Mechanisms	Main idea	Advantage	Disadvantage
RTRM (Bai et al., 2013)	It is a dynamic replica management strategy based on response time.	• High performance • Low response time • High rapid data download • Low energy consumption • High availability	• Low reliability • Low load balancing • High replication cost
D2RS (Sun et al., 2012)	It is put forward with a brief survey of replication strategy suitable for distributed computing environments.	• High availability • Low bandwidth consumption • Low replication cost	• High user waiting time • Low-speed data access • Low load balancing
CDRM (Wei et al., 2010)	It captures the relationship between replica number and availability.	• High availability • Low bandwidth consumption • Low access cost • High load balancing	• Low reliability • High energy consumption • Low response time
CIR (Li et al., 2011)	It considers cost-effective data replication management as a purpose.	• High reliability • High availability • Low replication cost • Low energy consumption	• High response time • Low load balancing • High response time • Low load balancing
DCR2S (Gill and Singh, 2015)	It optimizes the cost of replication using the concept of knapsack problem.	• Low replication cost • High reliability • High availability	• Low consistency rates • Low load balancing • High response time
HQFR (J.-W. Lin et al., 2013)	It is the greedy algorithm that if the application has a higher QoS requirement.	• Low replication cost • High availability • High scalability	• High time complexity • High bandwidth consumption
Energy-efficient data replication (Boru et al., 2013)	In this replication approach, every data object is permanently stored at the CentralDB and depending on the access pattern, it is replicated in data center DB and Rack DBs.	• High reliability • High availability • Low bandwidth consumption • Low energy consumption	• High update rate • High replication cost

Furthermore, RFH as a high-efficient, fault-tolerant and suitable mechanism for global wide replication employs traffic load evaluation to figure out the nodes that are in the traffic hub to solve flash crowd problem. MORM as one of the energy effective static approach outperforms default replication management of HDFS and MOE system in terms of load balancing and performance for large-scale cloud storage cluster. It seeks the near optimal solutions by balancing the trade-offs among the mean service time, load variance, mean file unavailability, energy consumption and mean access latency. MOE considered an evolutionary way to find the optimal replication strategy. It offers better scalability and performance, improves storage costs, access latency, and data availability. The optimal metadata method improves load balancing and it guarantees to achieve near optimal solution for MDS. Finally, SSOR considered the consistency in service oriented and cloud-based applications.

In dynamic replication strategies, the placement of the replicas are changed dynamically according to the environment and a number of replicas are not constant, therefore, when replicas are created, how many replicas are created, and the location of replica are very challenges. RTRM as first discussed mechanism in this category presents a dynamic replica management strategy based on response time which consists of replica creation, selection, and placement mechanisms. It sets a threshold for response time and does not consider load balancing. On the other hand, D2RS is a dynamic data replication approach that is suitable for distributed computing environments. It increases data availability, improves cloud system task successful execution rate and minimizes cloud system bandwidth consumption. On the other hand, CDRM further places replicas among cloud nodes to minimize blocking probability to improve load balance and overall performance and is implemented on HDFS. Besides, it maintains a rational number of replica, which not only satisfies availability, but also optimizes load balance, access latency, and keeps the whole storage system stable. Furthermore, CDRM captures the relationship between availability and replica number. CIR applies an incremental replication approach to optimize the number of replicas while meeting the reliability requirement in order to reach the cost-effective data replication management goal. However, it does not consider the issue of the trade-offs between cost and performance. DCR2S improves the cost of replication using the concept of knapsack problem, achieves high system byte effective rate and is effective in heterogeneous cloud system architecture. Finally, HQFR is a greedy algorithm and it cannot discover the optimal solution to the QADR problem. Finally, the energy efficient method improves performance metrics such as availability of network bandwidth, optimizes the energy efficiency of the system and optimization of communication delays in the quality of user experience of cloud applications. To side-by-side overview and compare all the reviewed mechanisms, Table 4 shows their main features.

4. Open issue

From this review, it can be obtained that there are still a lot of works to be prepared in the field of data replication in data cloud environment. Therefore, some open important research problems are discussed in this section.

It has been observed that there has not been a standard architecture for data replication in a cloud environment. Most of the discussed papers used a hierarchal architecture, but actually, a general graph is a more realistic architecture. So, the modifications of the hierarchal architectures to make it closer to the real cloud environment is very interesting. Also, it has been observed that there is no single strategy that addresses all issues involved in data replication. For example, some strategies consider improving reliability, scalability, and fault tolerance, as well as reducing the user waiting time, speeding up data access and enhancing the load balancing while some totally ignore some of these issues. Some approaches consider that conserving the network bandwidth is more important while some strategies have used more bandwidth

Table 3
Features of analyzed replication techniques in the paper.

Category	Static							Dynamic							
Approaches	**RFH**	**GFS**	**MOE**	**MinCopySet**	**MORM**	**MORG**	**Optimal meta-data**	**SSOR**	**D2RS**	**CDRM**	**CIR**	**RTRM**	**DCR2S**	**HQFR**	**Energy effective**
Year	2012	2003	2009	2013	2013	2009	2014	2013	2012	2010	2011	2013	2015	2013	2015
Availability	★★★	★★★	★	★★★	★★★	★★★	★★	★★★	★★★	★★★	★★★	★★★	★★★	★★★	★★★
Response time	★	★★★	★	★	★★★	★	★★★	★★	★	★	★	★★★	★	★★	★★
Reliability	★★★	★★★	★	★	★★★	★★★	★★	★★	★★★	★	★★★	★	★★★	★★★	★★★
Bandwidth consumption	★★★	★	★	★	★	★	★★	★★	★★★	★★★	★★★	★★★	★★★	★	★★★
Load balancing	★★★	★★★	★★★	★★★	★★	★★	★★★	★★★	★★★	★★★	★	★	★	★★	★★
Access cost	★★	★	★	★★★	★★	★	★★	★	★★★	★★★	★★★	★	★★★	★★	★★
Replication cost	★★★	★	★★	★★	★★	★	★	★	★★★	★★★	★★★	★	★★★	★★	★
Storage cost	★★★	★	★★★	★★★	★★★	★	★	★	★★★	★★	★★★	★	★★	★★★	★
Energy consumption	★	★	★	★	★★★	★	★★	★★	★★★	★	★★★	★★★	★★	★★	★★★
Consistency consideration	No	No	No	No	No	No	No	Yes	No	Yes	No	No	No	Yes	No

than average. Therefore, designing comprehensive method to consider the important parameters of data replication problem in a cloud environment is a very challenging.

Furthermore, most of the techniques included in this survey have used simulation to evaluate and test their mechanisms and algorithms. As a next step, these techniques must be prototyped and tested in the real world scenarios and also in scientific clouds. Therefore, providing the realistic evaluation of the assumptions that have been made for the reviewed mechanisms is very interesting. In addition, evaluating the mechanisms for larger task sets, combining both simulation and live environments, using other data-intensive applications is an interesting line for future research. With more realistic configurations, we expect that considering locality of access patterns can result in many advantageous.

It is also observed from this survey that some of the reviewed data replication mechanisms just explore a number of fix replica to achieve better performance, these mechanisms can be extended to dynamic data replication. Choosing and accessing to appropriate data resource are very important to optimize the use of cloud resource. They depend on different parameters including network status, characteristics of transfer, and replica host load. Therefore, the file replica selection process must choose a replica for jobs and users based on these parameters. Choosing a replica that provides the best performance can be considered as an interesting line for future research.

Also, trying to further reduce the job execution time is still interesting. There are two factors to be considered in decreasing the job execution time including the length of a time interval and exponential decay. In the length of a time interval, if the length is too short, the information about data access history is not enough. On the contrary, the information could be overdue and useless if the length is too long. Also, most of the discussed mechanisms considered the data in a cloud environment is read-only and hence replication strategy is very simple. However, in actual environments, the data is not always read only; rather it is updateable. Therefore, the replication approaches are unable to handle the consistency of data. Therefore, one of the important challenges of data replication is consistency between replicas. So, introducing a consistency protocols in order to present high consistency rates are very interesting. Also, the effect of different replica consistency algorithms on the overall performance of the cloud can be considered as a future work. Furthermore, exploring different adaptive replication algorithms that dynamically select replication algorithms depending on current conditions can be very interesting.

Re-examination of the multiple location mechanisms by considering more realistic constraints as well as seeking for better approximations can be considered another line for future research. In addition, implementing the discussed mechanisms in a service-oriented cloud environment is very interesting. As another part of future work, the discussed mechanisms can be extended to parallel applications where jobs may have precedence constraints and communicate with each other during their executions. The next step in the future can be combining the discussed approaches with various job scheduling policies, in particular, file-location-aware job scheduling. Furthermore, developing an implementation and verification of the reviewed solutions are another interesting future work.

Finally, it is very interesting to improve the replica selection process by involving the users in determining their preferences. Some users believe that the security is the most important factor while others may believe the reliability has the most importance. Thus, creating another component to provide searching and matching services for the users in the discussed mechanisms are very challenging. Still, expanding the discussed mechanisms and proposing a new replication strategy that supports replica management in terms of replica creation, deletion, and placement, to reduce both job execution time and network traffic can be considered as another line for future research. Finally, deploying the fairness concept and method into the discussed mechanisms is still an interesting line for future work.

5. Conclusion

In this paper, we have reviewed the past and the state of the art mechanisms in the field of cloud data replication. Furthermore, we introduced a taxonomy of the reviewed cloud data replication mechanisms. We also have divided the cloud data replication mechanisms into two main categories: static and dynamic. Static approaches determine the locations of replication nodes during the design phase while dynamic ones select replication nodes at a run time. Some dynamic approaches even allow their associated replication strategies to be adjusted at run time according to changes in user behavior and network topology. Dynamic replication is generally more appropriate for a service-oriented environment where the number and location of the users who intend to access data often have to be determined in a highly dynamic fashion. For each of these classes, we reviewed and compared several proposed mechanisms. In the static replication strategy, the number of replicas and their locations is initially set in advance. Instead, dynamic replication strategy dynamically creates and deletes replicas according to changing environment load conditions. There has been an interesting number of works for

data replication in the Cloud computing. Where most of them compared and analyzed in this paper. Also, by comparing the discussed mechanisms, As far as we know there is not a new method. Therefore, a specific mechanism to provide all mentioned issues will become a challenging problem and are interesting lines for future research and work.

Appendix

This section introduces the basic concepts and related terminologies which are used in the field of data replication. We explain the following concepts and terminologies:

Availability: It refers to property that a system is ready to be used immediately. In general, it refers to the probability that the system is operating properly at any given moment and is available to perform its functions on behalf of its users. In other words, a highly available system is one that will most likely be working at a given instant in time (Tanenbaum and Van Steen, 2007).

Bandwidth consumption: It describes the maximum data transfer rate of a network. The bandwidth consumption is typically low in the uplink where is used for sending queries and update requests. However, they are sent only at the fraction of the access rate. In the downlink, the required bandwidth is mainly determined by the size of the data items and the data access rate (Boru et al., 2013).

Consistency: Consistency is an important issue in any replication strategy. In any replication mechanisms having multiple copies may lead to consistency problems. When a copy is altered, that copy becomes different from the rest. So, changes have to be carried out on all copies to ensure consistency. Exactly when and how those changes need to be carried out determines the price of the replication (Tanenbaum and Van Steen, 2007).

Energy consumption: The energy consumed by the computing servers as well as core, aggregation, and access switches (Boru et al., 2013).

Reliability: It refers to the property that a system can run uninterruptedly without failure. In contrast to availability, reliability is defined in terms of a time interval instead of an instant in time. A highly-reliable system is one that will most likely continue to work without a disruption during a relatively long period of time.

Response time: It is the duration of time taken for a user or system to respond to a given stimulus or event. Bringing and maintaining the data closer to the servers where applications are executed significantly decrease access time for this data and greatly improves overall system performance. However, on the other side, the number and location of replicas should be selected carefully as excessive replication may increase the associated costs and traffic load in the data center network required for replica updates (Boru et al., 2013).

References

Aazami Ashkan, Ghandeharizadeh Shahram, Helmi Tooraj. Near optimal number of replicas for continuous media in ad-hoc networks of wireless devices. Paper presented at the Multimedia Information Systems; 2004.

Ahmad Noraziah, Fauzi Ainul Azila Che, Sidek Roslina Mohd, Zin Noriyani Mat, Beg Abul Hashem. Lowest data replication storage of binary vote assignment data grid. In: Proceedings of the networked digital technologies. Springer; 466–73.

Almorsy Mohamed, Grundy John, Ibrahim Amani S. Adaptable, model-driven security engineering for SaaS cloud-based applications. Autom Softw Eng 2014;*21*(2):187–224.

Ashouraie Mehran, Jafari Navimipour Nima, Ramage Magnus, Wong Patrick. Priority-based task scheduling on heterogeneous resources in the Expert Cloud. Kybernetes 2015;*44*(10).

Bai Xiaohu, Jin Hai, Liao Xiaofei, Shi Xuanhua, Shao Zhiyuan. RTRM: a response time-based replica management strategy for cloud storage system. In: Proceedings of the grid and pervasive computing. Springer; 124–33.

Berl Andreas, Gelenbe Erol, Di Girolamo, Marco Giuliani, Giovanni De Meer, Hermann Dang, Minh Quan, Pentikousis Kostas. Energy-efficient cloud computing. Comput J, *53*; 1045–51.

Bhardwaj Sushil, Jain Leena, Jain Sandeep. Cloud computing: a study of infrastructure as a service (IAAS). Int J Eng Inf Technol 2010;*2*(1):60–3.

Bjorkqvist M, Chen Lydia Y, Binder Walter. Optimizing service replication in clouds. Paper presented at the Simulation Conference (WSC), Proceedings of the 2011 Winter; 2011.

Bonvin Nicolas, Papaioannou Thanasis G, Aberer Karl. Dynamic cost-efficient replication in data clouds. Paper presented at the Proceedings of the 1st workshop on automated control for datacenters and clouds; 2009.

Boru Dejene, Kliazovich Dzmitry, Granelli Fabrizio, Bouvry Pascal, Zomaya Albert Y. Energy-efficient data replication in cloud computing datacenters. Paper presented at the Globecom Workshops (GC Wkshps), 2013 IEEE; 2013.

Buxmann Peter, Hess Thomas, Lehmann Sonja. Software as a Service. Wirtschaftsinformatik 2008;*50*(6):500–3.

Buyya Rajkumar, Ranjan Rajiv. Special section: federated resource management in grid and cloud computing systems. Futur Gener Comput Syst 2010;*26*(8):1189–91.

Buyya Rajkumar, Yeo Chee Shin, Venugopal Srikumar. Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities. Paper presented at the 10th IEEE international conference on High Performance Computing and Communications, 2008. HPCC'08.

Chang Ruay-Shiung, Chang Hui-Ping. A dynamic data replication strategy using access-weights in data grids. J Supercomput 2008;*45*(3):277–95.

Chen Tao, Bahsoon Rami, Tawil Abdel-Rahman H. Scalable service-oriented replication with flexible consistency guarantee in the cloud. Inf Sci 2014;*264*:349–70.

Chen Yiyu, Das Amitayu, Qin Wubi, Sivasubramaniam Anand, Wang Qian, Gautam Natarajan. Managing server energy and operational costs in hosting centers. Paper presented at the ACM SIGMETRICS performance evaluation review; 2005.

Choudhary Vidyanand. Software as a service: Implications for investment in software development. Paper presented at the 40th annual Hawaii International Conference on System Sciences, 2007. HICSS 2007.

Cidon, Asaf, Stutsman, Ryan, Rumble, Stephen, Katti, Sachin, Ousterhout, John, Rosenblum, Mendel. MinCopysets: derandomizing replication in cloud storage. Paper presented at the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI).

Dabrowski Christopher. Reliability in grid computing systems. Concurr Comput: Pract Exp 2009;*21*(8):927–59.

Dayyani Sheida, Khayyambashi Mohammad Reza. A comparative study of replication techniques in grid computing systems; 2013 [arXiv preprint arXiv:1309.6723].

Dinesha HA, Agrawal VK. Multi-level authentication technique for accessing cloud services. Paper presented at the 2012 International Conference on Computing, Communication and Applications (ICCCA).

Doğan Atakan. A study on performance of dynamic file replication algorithms for real-time file access in data grids. Futur Gener Comput Syst 2009;*25*(8):829–39.

Eludiora Safiriyu, Abiona Olatunde, Oluwatope Ayodeji, Oluwaranti Adeniran, Onime Clement, Kehinde Lawrence. A user identity management protocol for cloud computing paradigm. Int J Commun Netw Syst Sci 2011;*4*(03):152.

Ghemawat Sanjay, Gobioff Howard, Leung Shun-Tak. The Google file system. Paper presented at the ACM SIGOPS operating systems review; 2003.

Gill Navneet Kaur, Singh Sarbjeet. Dynamic cost-aware re-replication and rebalancing strategy in cloud system. Paper presented at the Proceedings of the 3rd international conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014; 2015.

Goel Sushant, Buyya Rajkumar. Data replication strategies in wide area distributed systems. Enterp Serv Comput: Concept Deploy 2006:*17*.

Hassan Osama Al-Haj, Ramaswamy Lakshmish, Miller John, Rasheed Khaled, Canfield E Rodney. Replication in overlay networks: a multi-objective optimization approach. In: Collaborative computing: networking, applications and worksharing. Springer; 2009. p. 512–28.

He Jing, Zhang Yanchun, Huang Guangyan, Shi Yong, Cao Jie. Distributed data possession checking for securing multiple replicas in geographically-dispersed clouds. J Comput Syst Sci 2012;*78*(5):1345–58.

Intanagonwiwat Chalermek, Govindan Ramesh, Estrin Deborah. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. Paper presented at the Proceedings of the 6th annual international conference on Mobile computing and networking; 2000.

Iosup Alexandru, Prodan Radu, Epema Dick. Iaas cloud benchmarking: approaches, challenges, and experiences. In: Cloud Computing for Data-Intensive Applications. Springer; 2014. p. 83–104.

Jafari Navimipour Nima, Masoud Rahmani Amir, Habibizad Navin Ahmad, Hosseinzadeh Mehdi. Job scheduling in the Expert Cloud based on genetic algorithms. Kybernetes 2014;*43*(8):1262–75.

Jin Shudong, Wang Limin. Content and service replication strategies in multi-hop wireless mesh networks. Paper presented at the proceedings of the 8th ACM international symposium on modeling, analysis and simulation of wireless and mobile systems; 2005.

Khajeh-Hosseini Ali, Greenwood David, Sommerville Ian. (2010). Cloud migration: A case study of migrating an enterprise it system to IaaS. Paper presented at the 2010 IEEE 3rd international conference on.Cloud Computing (CLOUD); 2010.

Lamehamedi Houda, Szymanski Boleslaw K. Decentralized data management framework for data grids. Future Gener Comput Syst 2007;*23*(1):109–15.

Lee Lin-Wen, Scheuermann Peter, Vingralek Radek. File assignment in parallel I/O systems with minimal variance of service time. IEEE Trans Comput 2000;*49*(2):127–40.

Lei Ming, Vrbsky, Susan V, Hong Xiaoyan. An on-line replication strategy to increase availability in data grids. Future Gener Comput Syst 2008;*24*(2):85–98.

Li Wenhao, Yang Yun, Yuan Dong. A novel cost-effective dynamic data replication strategy for reliability in cloud data centres. Paper presented at the 2011 IEEE ninth international conference on Dependable, Autonomic and Secure Computing (DASC); 2011.

Lin Jenn-Wei, Chen Chien-Hung, Chang J Morris. QoS-aware data replication for data-intensive applications in cloud computing systems. IEEE Trans Cloud Comput 2013;*1*(1):101–15.

Lin Jimmy, Fu D, Zhu J. What is cloud computing? IT a Serv 2009;*11*(2):10.

Lin Minghong, Wierman Adam, Andrew Lachlan LH, Thereska Eno. Dynamic right-sizing for power-proportional data centers. IEEE/ACM Trans Netw 2013;*21*(5):1378–91.

Long Sai-Qin, Zhao Yue-Long, Chen Wei. MORM: a Multi-objective Optimized Replication Management strategy for cloud storage cluster. *J Syst Arch* 2013.

Long Sai-Qin, Zhao Yue-Long, Chen Wei. MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster. J Syst Arch 2014;*60*(2):234–44.

Mell Peter, Grance Tim. Draft NIST working definition of cloud computing. Referenced on June 3, 2009. p. 15.

Meroufel Bakhta, Belalem Ghalem. Managing data replication and placement based on availability. AASRI Procedia 2013;*5*:147–55.

Miller Michael, Lei J. Cloud computing. *M. Beijing*: *Machine Industry Publication*; *4*.

Nathani Amit, Chaudhary Sanjay, Somani Gaurav. Policy based resource allocation in IaaS cloud. Future Gener Comput Syst 2012;*28*(1):94–103.

Navimipour N Jafari, Milani F Sharifi. Task scheduling in the cloud computing based on the cuckoo search algorithm. Int J Model Optim 2015;*5*(1):44.

Navimipour Nima Jafari. A formal approach for the specification and verification of a trustworthy human resource discovery mechanism in the Expert Cloud. *Expert Syst Appl* 2015.

Navimipour Nima Jafari, Milani Farnaz Sharifi. A comprehensive study of the resource discovery techniques in Peer-to-Peer networks. Peer-to-Peer Netw Appl 2014;*8*(3):474–92.

Navimipour Nima Jafari, Navin Ahmad Habibizad, Rahmani Amir Masoud, Hosseinzadeh Mehdi. Behavioral modeling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources. *Comput Ind* 2015a.

Navimipour Nima Jafari, Rahmani Amir Masoud, Navin Ahmad Habibizad, Hosseinzadeh Mehdi. Resource discovery mechanisms in grid systems: a survey. J Netw Comput Appl 2014;*41*:389–410.

Navimipour Nima Jafari, Rahmani Amir Masoud, Navin Ahmad Habibizad, Hosseinzadeh Mehdi. Expert cloud: a cloud-based framework to share the knowledge and skills of human resources. Comput Human Behav 2015b;*46*:57–74.

Navin Ahmad Habibizad, Navimipour Nima Jafari, Rahmani Amir Masoud, Hosseinzadeh Mehdi. Expert grid: new type of grid to manage the human resources and study the effectiveness of its task scheduler. Arab J Sci Eng 2014;*39*(8):6175–88.

Oussalah Mourad, Professor Ali Hessami Dr, Jafari Navimipour Nima, Masoud Rahmani Amir, Habibizad Navin Ahmad, Hosseinzadeh Mehdi. Job scheduling in the Expert Cloud based on genetic algorithms. Kybernetes 2014;*43*(8):1262–75.

Qiu Lili, Padmanabhan Venkata N, Voelker Geoffrey M. On the placement of web server replicas. Paper presented at the proceedings of twentieth annual joint conference of the IEEE Computer and Communications Societies. INFOCOM 2001. IEEE; 2001.

Qu Yanzhen, Xiong Naixue. RFH: a resilient, fault-tolerant and high-efficient replication algorithm for distributed cloud storage. Paper presented at the 2012 41st International Conference on Parallel Processing (ICPP); 2012.

Rahman Rashedur M, Barker Ken, Alhajj Reda. Replica placement design with static optimality and dynamic maintainability. Paper presented at the sixth IEEE international symposium on Cluster Computing and the Grid, 2006. CCGRID 06; 2006.

Sellami Mohamed, Yangui Sami, Mohamed Mohamed, Tata Samir. PaaS-independent provisioning and management of applications in the cloud. Paper presented at the 2013 IEEE sixth international conference on Cloud Computing (CLOUD); 2013.

Shvachko Konstantin, Kuang Hairong, Radia Sanjay, Chansler Robert. The hadoop distributed file system. Paper presented at the 2010 IEEE 26th symposium on Mass Storage Systems and Technologies (MSST); 2010.

Souri Alireza, Navimipour Nima Jafari. Behavioral modeling and formal verification of a resource discovery approach in Grid computing. Expert Syst Appl 2014;*41*(8):3831–49.

Sun Da-Wei, Chang Gui-Ran, Gao Shang, Jin Li-Zhong, Wang Xing-Wei. Modeling a dynamic data replication strategy to increase system availability in cloud computing environments. J Comput Sci Technol 2012;*27*(2):256–72.

Talia Domenico, Trunfio Paolo, Marozzo Fabrizio. Introduction to Cloud Computing. In: Marozzo DTT, editor. Data analysis in the cloud. Boston: Elsevier; 2016. p. 27–43 [chapter 2].

Tanenbaum Andrew, Van Steen Maarten. Distributed systems. Pearson Prentice-Hall; 2007.

Tang Bin, Gupta Himanshu, Das Samir R. Benefit-based data caching in ad hoc networks. IEEE Trans Mob Comput 2008;*7*(3):289–304.

Wang Wei, Liang Ben, Li Baochun. Revenue maximization with dynamic auctions in iaas cloud markets. Paper presented at the 2013 IEEE/ACM 21st International Symposium on Quality of Service (IWQoS); 2013.

Wei Qingsong, Veeravalli Bharadwaj, Gong Bozhao, Zeng Lingfang, Feng Dan. CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster. Paper presented at the 2010 IEEE International Conference on Cluster Computing (CLUSTER); 2010.

Wolfson Ouri, Jajodia Sushil, Huang Yixiu. An adaptive data replication algorithm. ACM Trans Database Syst 1997;*22*(2):255–314.

Xie Tao, Sun Yao. A file assignment strategy independent of workload characteristic assumptions. ACM Trans Storage (TOS) 2009;*5*(3):10.

Zeginis Dimitris, D'Andria Francesco, Bocconi, Stefano Gorronogoitia Cruz, Jesus Collell Martin, Oriol Gouvas, Panagiotis Tarabanis, Konstantinos A. A user-centric multi-PaaS application management solution for hybrid multi-cloud scenarios. Scalable Comput: Pract Exp 2013;*14*(1).

Zeng Zeng, Veeravalli Bharadwaj. Optimal metadata replications and request balancing strategy on cloud data centers. J Parallel Distrib Comput 2014;*74*(10):2934–40.