ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

A Survey on Dynamic Data Replication System in Cloud Computing

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

Department of CSE, PRIST University, Vallam, Thanjavur, India[1]

Department of CSE, Velammal Institute of Technology, Panchetti, Chennai, India[2]

ABSTRACT: Cloud computing has been a huge-scale parallel with shared computing system. It comprises an accumulation of inter-related plus virtualized calculating resources which are managed to be one or more amalgamated calculating resources. The target of the survey is a mechanized evaluation or else explanation of proceeding events in cloud data replication system. Completely, 34 papers are accumulated deriving of the standard publication IEEE. Every paper contains different techniques of data replication. According to the replication process, the papers are classified as four groups and the four groups are Variant kinds of replication systems, Survey on data replication systems, Data placement on data replication systems and also Fault tolerance on data replication systems. Analyzing the accomplishment of these promulgated tasks is calculated effectively with the analysis metrics of output, Accomplishment time, Storage usage, Replica number, Replication cost Recovery time, Network usage and Hit ratio. Higher output value denotes that the suggested method attains improved outcome. Thus our paper assists for enhancing novel technologies through the data replication system.

KEYWORDS: Throughput, Execution time, Storage utilization, Replication cost, Replica number, Recovery time ,Network usage, and Hit ratio.

I. INTRODUCTION

Cloud computing is an emerging practice that offers more flexibility in infrastructure and reduces cost than our traditional computing models. Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet) [6]. Cloud Computing is a term used to describe both a platform and type of application. As a platform it supplies, configures and reconfigures servers, while the servers can be physical machines or virtual machines.

Cloud computing consists of a collection of inter-connected and virtualized computing resources that are managed to be one or more unified computing resources. Further, the provided abstract, virtual resources, such as networks, servers, storage, applications and data, can be delivered as a service rather than a product.[10]. Google is realistic heterogeneous cloud computing, which provides different infrastructure along with resource types to satisfied the users requirements [5]. The big and giant web based companies such as Google, Face book, Twitter, Amazon, Salesforce.com came with a model named "Cloud Computing " the sharing of web infrastructure to deal with the internet data storage, scalability and computation [11]. Services are delivered on demand to the end-users over high-speed Internet as three types of computing architecture, namely Software as a Service (SAAS), Platforms as a Service (PAAS) and Infrastructure as a service (IAAS). The main goal is to provide users with more flexible services in a transparent manner, cheaper, scalable, highly available and powerful computing resources [10].

It mainly focuses on analyzing massive data. In order to analyze the dynamic, real-time and online data, the whole data cloud system must be improved in order to enhance the access speed and reliability and safety and system's load balance. Therefore, how to choose the replication strategy for data cloud is particular important. Creating replica is to reduce access latency and bandwidth consumption, in other words, it is to reduce the average job execution time and improve the usage of cloud resources [8]. File replication should provide fidelity of file consistency and unnecessary creation of replicas in various nodes, file replication should work in dynamic form that is it also allows peers to update, delete and provide consistent file in minimum partial tolerance [3]. It is well known that replication protocols perform differently depending on the workload characteristics. For instance, a read intensive partition may

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

provide a higher throughput with a primary-backup scheme [9]. In Cloud computing, replication is used for reducing user waiting time, increasing data availability and minimizing cloud system bandwidth consumption by offering the user multiple replicas of a specific service on different nodes [12]. The replication mechanism is divided into three important subjects: which file should be replicated, when to perform replication, and where the new replicas should be placed. Usually, replication from the server to the client is triggered when the popularity of a file passes a threshold and the client site is chosen either randomly or by selecting the least loaded site [1] [4].

Data grids deal with a huge amount of data regularly. It is a fundamental challenge to ensure efficient accesses to such widely distributed data sets. Creating replicas to a suitable site by data replication strategy can increase the system performance. It shortens the data access time and reduces bandwidth consumption [6]. Replicating data chunks at multiple clouds situated at geographically different locations would also have an additional decrease in response time [14]. However, as data sources and data processors integrated in a service application may be distributed geographically and connected with long-latency networks, data integration and sharing often lead to time and bandwidth penalties, thereby affecting the performance of the service application. This issue becomes more serious in large-scale, data intensive applications where large amounts of data have to be transported frequently between data sources and consumers. Data replication offers a practical solution to this issue by maintaining replicated copies of data in sites near to data consumers so as to reduce the time and bandwidth consumption of data transportation [2]. In addition, data replication can also improve the service performance and availability of data sources. Multiple replicated sites reduce the overhead imposed on a single point, and if one replicated site is not available, users can have access to the copies on other nodes [7]. The network environment is changeable, that makes the same replica sites are not always the best choices to download data to reduce the transmission time [13].

II. OVERVIEW OF DATA REPLICATION SYSTEM

Replication is used to advance system availability (by aiming traffic with a replica following a failure), prevent data reduction (by recovering lost files from a replica), and along with improve performance (by scattering load around multiple reproductions and by means of making low-latency access offered to users about the world). On the other hand, there are usually diverse ways to replication. Synchronous replication assures just about all copies are informed, but perhaps incurs excessive latency on updates. In addition, availability may be impacted in the event synchronously duplicated updates can't accomplish although some people might replicas are usually offline. Asynchronous replication excludes excessive write latency (in accurate, making the item appropriate pertaining to wide area replication) but permits replicas being stale. In addition, data loss normally takes place in the event an update is lost caused by breakdown just before it is usually replicated.

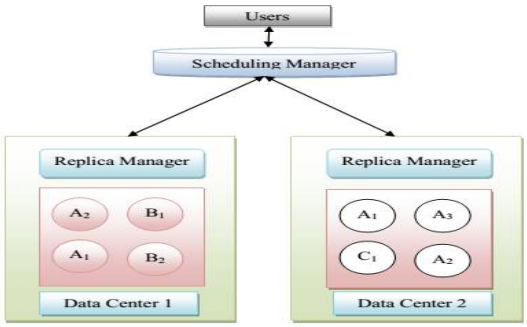

Fig. 1: Architecture of data replication system

Fig. 1 shows the architecture of the data replication system. The system architecture signifies the customers, scheduling manager, replica manager and data centers. Each and every main data centers are interlinked with each other and every single set of sub data centers are interlinked together with each other to be able to transfer your data from just one data center to one more. It isn't sure that every the data within the data center D1 could well be in your data center D2 and it isn't sure that the entire data within the sub data center SD1 of the main data center D1 is present in the sub data center SD2 of the main data center D1. In the event the user who uses your data center SD1 uses a data that's not within SD1

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

but that's present within SD2, your data center SD1 request a replica of the data to be able to SD2 as well as to its principal data center. Another event is that if a data within the data center SD1 is needed by more quantity of users and when a fresh user comes to use the identical data, user should wait to make use of that data. The task due to the end user is initial send for the scheduling manager. The scheduling manager then gives the path for the particular data center to gain access to the file. The path selection with the scheduling manager is based on the quantity of users' runs on the data center to stop congestion.

III. RELATED WORKS

What we have set out through these columns is to effectively furnish a general assessment of data replication system with a helping hand from the several investigational contributions in the literary world. On the whole we have selected as many as 40 study reports from several prominent publications like IEEE. These reports, in turn, are classified into 4 groups based on the replication process: Survey on data replication systems, Different types of replication systems, Data placement on data replication systems and Fault tolerance on data replication systems.

A. SURVEY ON DATA REPLICATION SYSTEM

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

Da-Wei Sun et al. [38] have put forward a dynamic data replication strategy with a brief survey of replication strategy suitable for distributed computing environments. It includes: 1) analyzing and modeling the relationship between system availability and the number of replicas; 2) evaluating and identifying the popular data and triggering a replication operation when the popularity data passes a dynamic threshold; 3) calculating a suitable number of copies to meet a reasonable system byte effective rate requirement and placing replicas among data nodes in a balanced way; 4) designing the dynamic data replication algorithm in a cloud. Experimental results have demonstrated the efficiency and effectiveness of the improved system brought by the proposed strategy in a cloud.

Optimistic replication techniques are different from traditional "pessimistic" ones. Instead of synchronous replica coordination, an optimistic algorithm propagates changes in the background, discovers conflicts after they happen and reaches agreement on the final contents incrementally. Yasushi Saito and Marc Shapiro [39] have explored the solution space for optimistic replication algorithms. They have identified the key challenges facing optimistic replication systems — ordering operations, detecting and resolving conflicts, propagating changes efficiently, and bounding replica divergence — and have provided a comprehensive survey of techniques developed for addressing these challenges.

B. DATA PLACEMENT TECHNIQUES FOR DATA REPLICATION

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

Takahiro Hara and Sanjay K. Madria [22] have solved the lower data accessibility problem by replicating data items on mobile hosts. They have proposed three replica allocation methods assuming that each data item was not updated. In those three methods, they have taken into account the access frequency from mobile hosts to each data item and the status of the network connection. Then, they have extended the proposed methods by considering a periodic updates and integrating user profiles consisting of mobile users' schedules, access behavior, and read/write patterns. They have also showed the results of simulation experiments regarding the performance evaluation of their proposed methods.

Gilbert et al. [41] have showed how a DTN-like messaging system can be readily built as a simple application on top of a peer-to-peer replication platform. In order to reduce delivery delays while retaining the desirable replication guarantees, they have then extended the replication substrate to permit pluggable DTN routing protocols. They have described the implementation of four representative DTN schemes as replication policies and evaluate these extensions

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

with emulations driven by traces of e-mail messaging and vehicular mobility. They have concluded that DTNs and replication systems can benefit substantially from a cross fertilization of ideas.

Kai Hwang and Deyi Li [46] have suggested using a trust-overlay network over multiple data centers to implement a reputation system for establishing trust between service providers and data owners. Data coloring and software watermarking techniques protect shared data objects and massively distributed software modules. These techniques safeguard multi-way authentications, enable single sign-on in the cloud, and tighten access control for sensitive data in both public and private clouds.

Bermbach et al. [50] have presented Meta Storage, a federated Cloud storage system that can integrate diverse Cloud storage providers. Meta Storage was a highly available and scalable distributed hash table that replicates data on top of diverse storage services. Meta Storage reuses mechanisms from Amazon's Dynamo for cross-provider replication and hence have introduced a novel approach to manage consistency latency tradeoffs by extending the traditional quorum (N, R, W) configurations to an (NP , R, W) scheme that have included different providers as an additional dimension. With Meta Storage, new means to control consistency-latency tradeoffs were introduced.

C. FAULT TOLERANCE ON DATA REPLICATION

Ravi Jhawar et al. [35] have introduced an innovative, system-level, modular perspective on creating and managing fault tolerance in Clouds. They have proposed a comprehensive high-level approach to shading the implementation details of the fault tolerance techniques to application developers and users by means of a dedicated service layer. In particular, the service layer allows the user to specify and apply the desired level of fault tolerance, and does not require knowledge about the fault tolerance techniques that are available in the envisioned Cloud and their implementations.

Yuriy Brun and Nenad Medvidovic [36] have addressed the problem of distributing computation onto the cloud in a way that preserves the privacy of the computation's data even from the cloud nodes themselves. The approach, called sTile, separates the computation into small sub computations and distributes them in a way that makes it prohibitively hard to reconstruct the data. They have evaluated sTile theoretically and empirically: First, they have proved that sTile systems preserve privacy. Second, they have done a prototype implementation on three different networks, including the globally-distributed Planet Lab testbed, in order to show that sTile was robust to network delay and efficient enough to significantly outperform existing privacy-preserving approaches.

Kehuan Zhang et al. [42] have presented a suite of new techniques that make such privacy-aware data-intensive computing possible. Their system was called Sedic, leverages the special features of MapReduce to automatically partition a computing job according to the security levels of the data it works on, and arranges the computation across a hybrid cloud. Specifically, they have modified MapReduce's distributed file system to strategically replicate data, moving sanitized data blocks to the public cloud. Over this data placement, map tasks were carefully scheduled to outsource as much workload to the public cloud as possible, given sensitive data always stay on the private cloud. In order to minimize inter-cloud communication, their approach also have automatically analyzed and transforms the reduction structure of a submitted job to aggregate the map outcomes within the public cloud before sending the result back to the private cloud for the final reduction. This also allows the users to interact with our system in the same way they work with MapReduce, and directly run their legacy code in the framework. They implemented Sedic on Hadoop and evaluated it using both real and synthesized computing jobs on a large-scale cloud test-bed. The study shows that our techniques effectively protect sensitive user data, offload a large amount of computation to the public cloud and also fully preserve the scalability of MapReduce.

To continuously support the QoS requirement of an application after data corruption, Jenn-Wei Lin et al. [43] have proposed two QoS-aware data replication (QADR) algorithms in cloud computing systems. The first algorithm adopts the intuitive idea of high-QoS first-replication (HQFR) to perform data replication. However, this greedy algorithm cannot minimize the data replication cost and the number of QoS-violated data replicas. To achieve these two minimum objectives, the second algorithm transformed the QADR problem into the well-known minimum-cost maximum-flow (MCMF) problem. By applying the existing MCMF algorithm to solve the QADR problem, the second algorithm can produce the optimal solution to the QADR problem in polynomial time, but it takes more computational time than the first algorithm. Moreover, it was known that a cloud computing system usually has a large number of nodes. They have also proposed node combination techniques to reduce the possibly large data replication time. Finally, simulation experiments were performed to demonstrate the effectiveness of the proposed algorithms in the data replication and recovery.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

George V. Popescu and Christopher F. Codella [47] have presented Quality of Service (QoS) architecture for just-in-time data replication in network Virtual Environments. Quality of service was achieved by predicting the load and adapting to network traffic variations. Data was prefetched at the client based on network traffic estimates and viewpoint navigation prediction. QoS negotiation allows the server to control the network resources allocated per client. Experimental results have showed that QoS data replication can be implemented with reasonably small network and server overload.

D. DYNAMIC DATA REPLICATION TECHNIQUES

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

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

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

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

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

Chao-Tung Yang et al. [23] have proposed a dynamical maintenance service of replication to maintain the data in grid environment. Replicas were adjusted to the appropriate location for usage. The BHR algorithm was a strategy to maintain replica dynamically. We have pointed out a scenario that the BHR algorithm will cause a mistake when it operates. The above said mistake limits the speed of transferring file. They have proposed the Dynamic Maintenance Service which was aimed at overcoming the mistake proposed. The contributions of their proposed work are that the data grid environment provides more efficiency by using DMS algorithm. The experimental results have showed that the DMS algorithm was more efficient than other replication strategies.

Data Grid provides geographically distributed storage resources for large-scale data-intensive applications that generate large data sets. Because data was the important resource in data grids, an efficient management was needed to

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

minimize the response time of applications. Replication was typically used in data grids to improve access time and to reduce the bandwidth consumption. Faouzi Ben Charrada et al. [24] have proposed a new replication strategy for dynamic data grids which takes into account the dynamicity of sites. Indeed, the dynamicity of sites was an important challenge in grids. Their strategy had helped to increase file availability, in order to improve the response time and to reduce the bandwidth consumption. They had evaluated their strategy through simulation using OptorSim. The results they have obtained appeared to be promising.

Gao Gai-Mei and Bai Shang-Wang [25] have discussed the dynamic replication strategies for the data grids. They have included the intra-domain derivative strategy and the cross-domain expansion strategy. A cross-domain expansion strategy of cascading replication plus economic model was described which can evaluate the performance of three different kind access patterns by making use of three different replication strategies. The stimulation results obtained have showed that the replication strategy have significantly reduced the average working time and have improved the utilization of grid resources and have validated the model's superiority.

M. Zarina et al. [26] have proposed a model of replication strategy in federated data grid that was known as dynamic replication for federation (DRF). DRF have used the concept of a defined 'network core area' (NCA) as the designated search area of which a search was focused. However, the area known as NCA was not static and it was bound to change if data requested cannot be found. They also have highlighted how NCA was defined and reallocated if a search fails. Results from the analysis have showed that the DRF proposed was superior to Optimal Downloading Replication Strategy (ODRS) in terms of wide area network bandwidth requirement and in the average access latency data.

Wenhao LI et al. [28] have proposed a novel cost-effective dynamic data replication strategy which facilitates an incremental replication method to reduce the storage cost and meet the data reliability requirement at the same time. This replication strategy has worked very well especially for data which were only used temporarily and/or have a relatively low reliability requirement. The simulation have showed that their replication strategy for reliability can reduce the data storage cost in data centers substantially.

Ruay-Shiung Chang et al. [29] have proposed a dynamic data replication mechanism, which was called Latest Access Largest Weight (LALW). LALW selects a popular file for replication and calculates a suitable number of copies and grid sites for replication. By setting a different weight for each data access record, the importance of each record was differentiated. The data access records in the nearer past have higher weight. It has indicated that these records have higher value of references. A Grid simulator OptorSim was used to evaluate the performance of this dynamic replication strategy. The simulation results have showed that LAHW had successfully increased the effective network usage. It means that the LALW replication strategy can find out a popular file and replicates it to a suitable site.

Weizhong Lu et al. [23] have proposed a new data replication method called Hierarchical Replication Model (HRM). This method selects mobile hosts as data replicas holders taking into account not only data access frequencies and network topology, but also link bandwidth and remaining amount of batteries of hosts. This method groups the network into three hierarchies to improve the data accessibility, and it speeds up the data transmission and increase the data accessing efficiency at the same time. Furthermore, simulation results had verified the effectiveness of this method.

Zhendong Cheng et al. [31] have described ERMS, an elastic replication management system for HDFS. ERMS have provided an active/standby storage model for HDFS. It had utilized a complex event processing engine in order to distinguish real-time data types, and then had dynamically increased extra replicas for hot data, cleans up these extra replicas when the data cool down, and uses erasure codes for cold data. ERMS also had introduced a replica placement strategy for the extra replicas of hot data and erasure coding parities. The experiments had showed that ERMS effectively improved the reliability and performance of HDFS and had reduced the storage overhead.

The energy efficiency of the data centers storing this data was one of the biggest issues in data intensive computing. Since power was needed to transmit, store and cool the data, they had proposed to minimize the amount of data transmitted and stored by utilizing smart replication strategies that are data aware. Susan V. Vrbsky et al. [39] have presented a new data replication approach, called the sliding window replica strategy (SWIN) that was not only data aware, but was also energy efficient. They have measured the performance of SWIN and existing replica strategies on our Sage green cluster to study the power consumption of the strategies. Results from their study have implications beyond our cluster to the management of data in clouds.

Dharma Teja Nukarapu et al. [34] have proposed a data replication algorithm that not only has a provable theoretical performance guarantee, but also can be implemented in a distributed and practical manner. Specifically, they have designed a polynomial time centralized replication algorithm that had reduced the total data file access delay by at least half of that reduced by the optimal replication solution. Based on this centralized algorithm, they have also

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

designed a distributed caching algorithm, which can be easily adopted in a distributed environment such as Data Grids. Extensive simulations were performed to validate the efficiency of our proposed algorithms. Using the proposed simulator, they have showed that the centralized replication algorithm performs comparably to the optimal algorithm and other intuitive heuristics under different network parameters. Using GridSim, a popular distributed Grid simulator, they had demonstrated that the distributed caching technique significantly outperformed an existing popular file caching technique in Data Grids, and it was more scalable and adaptive to the dynamic change of file access patterns in Data Grids.

The CAP theorem explores tradeoffs between consistency, availability, and partition tolerance, and concludes that a replicated service can have just two of these three properties. To prove CAP, Kenneth P. Birman et al. [48] have constructed a scenario in which a replicated service was forced to respond to conflicting requests during a wide-area network outage, as might occur if two different datacenters hosted replicas of some single service, and received updates at a time when the network link between them was down. The replicas respond without discovering the conflict, resulting in inconsistency that might confuse an end user.

Zhou Wei et al. [44] have showed how one can support strict ACID transactions without compromising the scalability property of the cloud for Web applications. First, they load data from the cloud storage system into the transactional layer. Second, they split the data across any number of LTMs, and replicate them only for fault tolerance. Web applications typically access only a few partitions in any of their transactions, which gave CloudTPS linear scalability. CloudTPS supports full ACID properties even in the presence of server failures and network partitions. Recovering from a failure only causes a temporary drop in throughput and a few aborted transactions. Recovering from a network partition, however, may possibly cause temporary unavailability of CloudTPS, as they have explicitly chosen to maintain strong consistency over high availability. Their memory management mechanism can prevent LTM memory overflow. They have expected typical Web applications to exhibit strong data locality so this mechanism only introduces minor performance overhead. Data partitioning also implies that transactions can only access data by primary key. Read-only transactions that require more complex data access can still be executed, but on a possibly outdated snapshot of the database.

Tim Ho and David Abramson [45] have discussed a GriddLeS Replication Service (GRS) which allows existing applications to access replicated data. Importantly, this was performed without any modification to the application. They have described the architecture of the GRS, which provides a set of IO routines for access to different replica management systems, and have illustrated our implementation which only currently supports the SRB from SDSC. This abstraction avoids code modification because the application does not need to have specific code for any particular replica systems. In addition, the GRS provides certain level of optimization by dynamically switching to a better server. This was done by monitoring the network condition using the Network Weather Service at runtime and forecasting which connection has better bandwidth. The server selection can occur at any time during program execution to tackle unreliable network connection. This improves fault tolerance and provides a reliable data input because the same piece of data is available in several servers. They have demonstrated that by changing the data server based on the network condition can increase the overall performance.

IV. RESULTS AND DISCUSSIONS

Data Replication in cloud computing has been evaluated using techniques in our survey papers. In order to evaluate the performance results, initially all the papers are categorized based on Throughput, Execution time, Storage utilization, Replica number, Network usage, Replication cost, Recovery time and Hit ratio. The experimentations of these review papers are carried out in every paper with different techniques which are used in every survey papers are given in the following table I. It helps us to analyze the results of every method with detailed evaluation.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

Table 1: Various Techniques used in the survey papers

Reference numbers of papers	Methods used in each papers for cloud computing
[15]	Survey on Data Replication techniques
[16]	Dynamic replication strategy based on fast spread
[17]	Dynamic replication strategy based on two ideas
[18]	Matrix based K-Means clustering for data placement
[19]	Dynamic Data replication method PDDRA
[20]	Dynamic hierarchical replication algorithm
[21]	Adaptive data replication algorithm
[22]	Replica allocation Method
[23]	Dynamic maintenance service of replication
[24]	Replication strategy for dynamic data grids
[25]	Dynamic replication strategy for data grids
[26]	Dynamic replication for federation
[27]	Metrics for measuring system data availability
[28]	Cost-Effective dynamic data replication strategy
[29]	Dynamic data replication mechanism
[30]	Hierarchical replication model
[31]	Elastic Replication management system
[32]	Sliding window replica strategy
[33]	Polynomial time centralized replication algorithm
[34]	Fault tolerance management system
[35]	Privacy preservation approach sTile
[36]	Lazy replication algorithm
[37]	Survey on dynamic replication strategies
[38]	Survey on key challenges for optimistic replication systems
[39]	CAP Theorem
[40]	DTN-like messaging system to reduce the delivery delays in peer-to-peer replication system
[41]	Sedic for privacy aware data intensive computing
[42]	QOS aware data replication
[43]	ACID Transactions data replication
[44]	Trust overlay network over multiple data centers

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

[45]	QOS architecture
[46]	Dynamic multiple location replication
[47]	Article on cloud computing basics
[48]	Federated cloud storage system

Table 2: Performance measures analyzed on Various techniques

Replication Techniques used	PERFORMANCE MEASURES							
	Throughput	Execution time	Storage utilization	Replica number	Network usage	Replication cost	Recovery time	Hit ratio
Elastic replication Management	✓	✓	✓	✓				
Dynamic weighted data replication		✓	✓		✓			
QOS aware data replication						✓	✓	
Lazy database replication	✓	✓						
PDDRA		✓		✓	✓			✓

Table 1 shows the methods and algorithms that are used for the data replication in cloud computing. From the experimentation results of these reviewed papers, we have analyzed the performance of the various recognition systems. Some of the results of the experimentation with the evaluation values are shown below. In general, the following metrics such as Throughput, Execution time, Storage utilization, Replica number, Network usage, Replication cost, Recovery time and Hit ratio are used as the evaluation metrics in all data replication systems as given in Table 2. They are given in detail one by one with the comparison.

Effective Network Usage (ENU)

ENU is effectively the ratio of files transferred to files requested, so a low value indicates that the optimization strategy used is better at putting files in the right places. It ranges from 0 to 1.

$$ENU = \frac{\left(N_{remote\ file} + N_{file\ replications}\right)}{\left(N_{remote\ file} + N_{local\ file}\right)}$$

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

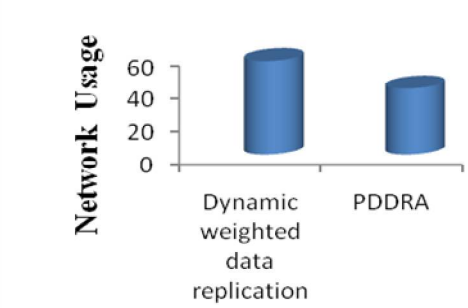

Fig. 2: Network usage values for various existing works for data replication

From the above fig 2, we can observe that the various methods have different Network usage values. Some of the methods from the survey works are taken for our evaluation work. The methods with the reference papers are Dynamic weighted data replication and PDDRA methods with the network usage values 57 and 40 respectively. Overall, we observe that the dynamic data replication techniques facilitate good performance results.

Hit ratio

Hit ratio is the ratio of Total number of Local File Accesses to all accesses containing Local File Accesses, Total number of Replications and Total number of Remote File Accesses.

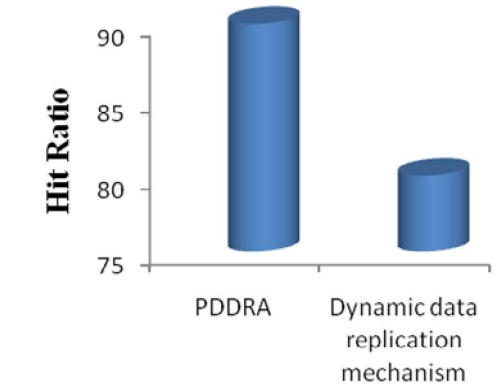

Fig. 3: Hit ratio values for various existing works for data replication

From the above fig 3, we can observe that the various methods have different Hit ratio values. Some of the methods from the survey works are taken for our evaluation work. The methods with the reference papers are Dynamic weighted data replication and PDDRA methods with the hit ratio values 80 and 90 respectively. Overall, we observe that the dynamic data replication techniques facilitate good performance results.

Total number of replications

Great number of replications shows that large numbers of files were not stored locally at the time of need, so replication was needed in order to access the required file.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

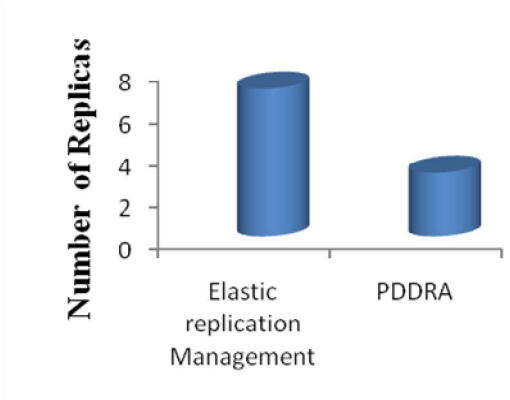

Fig. 4: Number of replications for various existing works for data replication

From the above fig 4, we can observe that the various methods have different number of replications. Some of the methods from the survey works are taken for our evaluation work. The methods with the reference papers are Elastic replication management and PDDRA methods with the replication number 7 and 3 respectively. Overall, we observe that the dynamic data replication techniques facilitate good performance results.

Replication cost

A replica with high replication cost is not a suitable candidate for replacement because if the grid site needs that replica in the future, it should pay a high cost for replicating it again, and this is not economical. Therefore, the RPV will be greater if Replication Cost of that replica is high.

$$\text{Re } plication \ \cos t = \frac{Size}{Bandwidth} * \text{Pr } opogation \ \ delay \ time$$

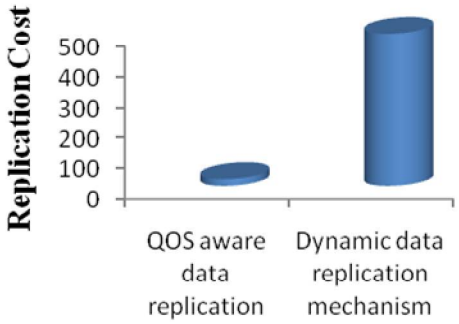

Fig. 5: Replication cost values for various existing works for data replication

From the above fig 5, we can observe that the various methods have different replication cost. Some of the methods from the survey works are taken for our evaluation work. The methods with the reference papers are Dynamic data replication and Qos aware data replication methods with the replication cost 500 and 22.23 respectively.

Execution Time

Execution time is the time taken for execution of the whole project. It will be calculated in seconds.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

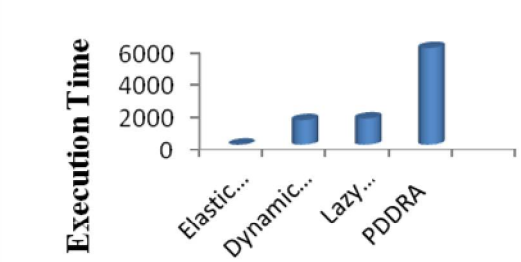

Fig. 6: Execution Time values for various existing works for data replication

From the above fig 6, we can observe that the various methods have different execution time. Some of the methods from the survey works are taken for our evaluation work. The methods with the reference papers are Elastic replication management, Dynamic weighted data replication, lazy database replication and PDDRA methods. Overall, we observe that the dynamic data replication techniques facilitate good performance results.

Throughput

Throughput refers to the performance of tasks by a computing service or device over a specific period. It measures the amount of completed work against time consumed and may be used to measure the performance of a process.

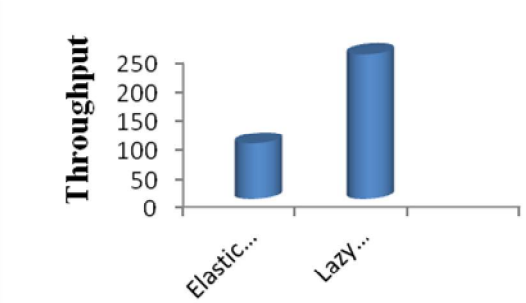

Fig. 7: Throughput values for various existing works for data replication

From the above fig 7, we can observe that the various methods have different throughput values. Some of the methods from the survey works are taken for our evaluation work. The methods with the reference papers are Elastic replication management and lazy database replication. Overall, we observe that the dynamic data replication techniques facilitate good performance results.

Storage Utilization

Storage Utilization is the amount of Giga Bytes utilized by the data in the cloud.

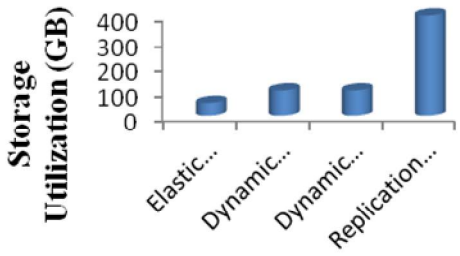

Fig. 8: Storage Utilization values for various existing works for data replication

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

From the above fig 8, we can observe that the various methods have different storage utilization values. Some of the methods from the survey works are taken for our evaluation work. The methods with the reference papers are Elastic replication management, Dynamic weighted data replication, and dynamic data replication mechanism and replication strategy for dynamic data grids. Overall, we observe that the dynamic data replication techniques facilitate good performance results.

V. CONCLUSION

Data replication is one of the popular tasks in cloud computing. In this paper, we have surveyed the topic of data replication. We have taken a total of 34 papers for our review work, in which the articles were taken from the standard publications. The overview of the data replication system in cloud computing is initially explained in a brief manner. After that we have categorized every paper under four groups based on the replication pattern. The performance of the data replication works has been extensively analyzed with the evaluation metrics – Throughput, Execution time, Replica number, Network usage, Storage utilization, Replication cost Recovery time and Hit ratio. From this evaluation, we observe that some of the methods have provided efficient performance result. Even though the performance of these methods has yielded good quality results, today's technology has been developing day by day, which calls for further improvement in the performance. So we have to update the new technologies every day. And we believe that a greater number of works will be developed in future by using our survey work.

REFERENCES

[1] Ruay-Shiung Chang, Hui-Ping Chang and Yun-Ting Wang, "A Dynamic Weighted Data Replication Strategy in Data Grids", In the proceedings of the 6th ACS/IEEE International Conference on Computer Systems and Applications, 2008.

[2] H. Stockinger, "Database Replication in Worldwide Distributed Data Grids", University of Vienna, 2001.

[3] Priya Deshpande, Aniket Bhaise and Prasanna Joeg, "A Comparative analysis of Data Replication Strategies and Consistency Maintenance in Distributed File Systems", International Journal of Recent Technology and Engineering (IJRTE), Vol. 2, 2013.

[4] Ruay-Shiung Chang and Hui-Ping Chang, "A dynamic data replication strategy using access-weights in data grids", Journal of Supercompute, 2008.

[5] Vijaya -Kumar-C and Dr. G.A. Ramachandhra, "Optimization of Large Data in Cloud computing using Replication Methods", International Journal of Computer Science and Information Technologies, Vol. 5, No. 3, pp. 3034-3038, 2014.

[6] Dheresh Soni, Atish Mishra, Satyendra Singh Thakur and Nishant Chaurasia, "Applying Frequent Pattern Mining in Cloud Computing Environment", International Journal of Advanced Computer Research, Vol. 1, No. 2, 2011.

[7] Feng Dan and Tan Zhipeng, "Dynamic replication strategies for object storage systems", In Proceeding of the 2006 international conference on Emerging Directions in Embedded and Ubiquitous Computing, pp. 53-61, 2006.

[8] A. M. Isa, A. N. M. M. Rose, M. Mat Deris and M. Zarina, "Dynamic data replication strategy based on federation data grid systems", In proceeding of the First international conference on Information computing and applications, pp. 25-32, 2010.

[9] Itziar Arrieta-Salinas, Jose Enrique Armend ariz-Inigo and Joan Navarro, "Classic Replication Techniques on the Cloud", In proceedings of the Seventh International Conference on Availability, Reliability and Security, 2012.

[10] Rajkumar Buyyaa, Chee Shin Yeoa, Srikumar Venugopala, James Broberga, Ivona Brandicc, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Vol. 25, No. 6, pp. 599–616, 2009.

[11] Sunil Kaushik, Pranjal Mishra and Dr Ashish Bharadwaj, "Critical Evaluation of Cloud Computing for Transactional and Analytical Databases", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, 2013.

[12] Mohamed-K Hussein and Mohamed-H Mousa, "A Light-weight Data Replication for Cloud Data Centers Environment", International Journal of Engineering and Innovative Technology (IJEIT), Vol. 1, No. 6, 2012.

[13] Bakhta Meroufel and Ghalem Belalem, "Dynamic Replication Based on Availability and Popularity in the Presence of Failures", Journal of Information Processing Systems, Vol.8, No.2, 2012.

[14] Veena Khandelwal, "Secure and Efficient Data Storage in Multi-clouds", International Journal of Information and Computation Technology, Vol. 3, No. 9, pp. 977-984, 2013.

[15] Tehmina Amjad, Muhammad Sher and Ali Daud, "A survey of dynamic replication strategies for improving data availability in data grids", Future Generation Computer Systems, Vol. 28, pp. 337–349, 2012.

[16] Mohammad Bsoul, Ahmad Al-Khasawneh, Yousef Kilani and Ibrahim Obeidat, "A threshold-based dynamic data replication strategy", The Journal of Supercomputing, Vol. 60, No. 3, pp. 301-310, 2012.

[17] Zhe Wang, Tao Li, Naixue Xiong and Yi Pan, "A novel dynamic network data replication scheme based on historical access record and proactive deletion", The Journal of Supercomputing, Vol. 62, No. 1, pp. 227-250, 2012.

[18] Dong Yuan, Yun Yang, Xiao Liu and Jinjun Chen, "A data placement strategy in scientific cloud workflows", Future Generation Computer Systems, Vol. 26, No. 8, pp. 1200–1214, 2010.

[19] Nazanin Saadat and Amir Masoud Rahmani, "PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids", Future Generation Computer Systems, Vol. 28, No. 4, pp. 666–681, 2012.

[20] Najme Mansouri and Gholam Hosein Dastghaibyfard, "A dynamic replica management strategy in data grid", Journal of Network and Computer Applications, Vol. 35, No. 4, pp. 1297–1303, 2012.

ISSN(Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2015

[21] Ming-Chang Leea, Fang-Yie Leub and Ying-ping Chen, "PFRF: An adaptive data replication algorithm based on star-topology data grids", Future Generation Computer Systems, Vol. 28, No. 7, pp. 1045–1057, 2012.
[22] Takahiro Hara and Sanjay K. Madria, "Data Replication for Improving Data Accessibility in ADHOC Networks", IEEE Transactions on Mobile Computing, Vol. 5, No. 11, 2006.
[23] Chao-Tung Yang, Chun-Pin Fu and Chien-Jung Huang, "A Dynamic File Replication Strategy in Data Grids", In proceedings of the IEEE International technical conference, pp. 1-5, 2007.
[24] Faouzi Ben Charrada, Habib Ounelli and Hanène Chettaoui, "An Efficient Replication Strategy for Dynamic Data Grids", In proceedings of IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2010.
[25] Gao Gai-Mei and Bai Shang-Wang, "Design and simulation of dynamic replication strategy for the data grid", In proceedings of the IEEE International Conference on Industrial control and electronic engineering, 2012.
[26] M. Zarina, M. Mat Deris, A.N.M.M. Rose and A.M. Is, "Dynamic Data Replication Strategy Based on Federation Data Grid Systems", In Proceedings of Springer first international conference on Information Computing and Applications, pp. 25-32, 2010.
[27] Ming Lei, Susan V. Vrbsky and Xiaoyan Hong, "A Dynamic Data Grid Replication Strategy to Minimize the Data Missed", In Proceedings of Third International Workshop on Networks for Grid Applications, 2006.
[28] Wenhao LI, Yun Yang and Dong Yuan, "A Novel Cost-effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centers", In Proceedings of the Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2011.
[29] Ruay-Shiung Chang, Hui-Ping Chang and Yun-Ting Wang,"A dynamic weighted data replication strategy in data grids, In proceedings of the 6th ACS/IEEE International Conference on Computer Systems and Applications, pp. 414-421, 2008.
[30] Weizhong Lu, Yuanchun Zhou, Jianhui Li and Baoping Yan, "A Hierarchical Data Replication Method in Ad Hoc Networks", In proceedings of the 2nd IEEE International Conference on Computer Engineering and Technology, vol. 7, pp. 23-27, 2010.
[31] Zhendong Cheng, Zhongzhi Luan, You Meng, Yijing Xu, Depei Qian, Alain Roy, Ning Zhang and Gang Guan, "ERMS : An Elastic Replication Management System for HDFS", In proceedings of IEEE International Conference on Cluster Computing Workshops, 2012.
[32] Susan V. Vrbsky, Ming Lei, Karl Smith and Jeff Byrd, "Data Replication and Power Consumption in Data Grids", In proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010.
[33] Dharma Teja Nukarapu, Bin Tang, Liqiang Wang,Member and Shiyong Lu, "Data Replication in Data Intensive Scientific Applications with Performance Guarantee", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 8, 2011.
[34] Ravi Jhawar, Vincenzo Piuri, and Marco Santambrogio, "Fault Tolerance Management in Cloud Computing: A System-Level Perspective", IEEE Systems Journal, Vol. 7, No. 2, pp. 288-297, 2013.
[35] Yuriy Brun and Nenad Medvidovic, "Keeping Data Private while Computing in the Cloud", In proceedings of IEEE Fifth International Conference on Cloud Computing, 2012.
[36] Khuzaima Daudjee and Kenneth Sale, "Lazy Database Replication with Ordering Guarantees", In Proceedings of the 20th IEEE International Conference on Data Engineering, 2004.
[37] Da-Wei Sun, Gui-Ran Chang, Shang Gao, Li-Zhong Jin and Xing-Wei Wang, "Modeling a Dynamic Data Replication Strategy to Increase System Availability in Cloud Computing Environments", Springer Journal of Computer Science and Technology, Vol. 27, No. 2, pp. 256–272, 2012.
[38] Yasushi Saito and Marc Shapiro, "Optimistic Replication", ACM Computing Surveys, Vol. 37, No. 1, pp. 42-81, 2005.
[39] Kenneth P. Birman, Daniel A. Freedman, Qi Huang and Patrick Dowell, "Overcoming CAP With Consistent Soft-State Replication", IEEE Computer Society, 2012.
[40] Peter Gilbert, Venugopalan Ramasubramanian, Patrick Stuedi and Doug Terry, "Peer-to-peer Data Replication Meets Delay Tolerant Networking", In proceedings of the 31st IEEE International Conference on Distributed Computing Systems, 2011.
[41] Kehuan Zhang, Xiaoyong Zhou, Yangyi Chen, XiaoFeng Wang and Yaoping Ruan, "Sedic: Privacy-Aware Data Intensive Computing on Hybrid Clouds", In Proceedings of the 18th ACM conference on Computer and communications security, pp. 515-526, 2011.
[42] Jenn-Wei Lin, Chien-Hung Chen and J. Morris Chang, "QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems", IEEE Transactions on Cloud Computing, Vol. 1, No. 1, 2013.
[43] Zhou Wei, Guillaume Pierre and Chi-Hung Chi, "CloudTPS: Scalable Transactions for Web Applications in the Cloud", IEEE Transactions on Services Computing, Special Issue on Cloud Computing, 2011.
[44] Kai Hwang and Deyi Li, "Trusted Cloud Computing with Secure Resources and Data Coloring", IEEE Computer Society, 2010.
[45] George V. Popescu and Christopher F. Codella, "An Architecture for QoS Data Replication in Network Virtual Environments", In Proceedings of the IEEE Virtual Reality, 2002.
[46] Xiaohua Dong, Ji Li, Zhongfu Wu, Dacheng Zhang and Jie Xu, "On Dynamic Replication Strategies in Data Service Grids", In proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing, pp. 155 – 161, 2008.
[47] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia, "A view of cloud computing", Communications of the ACM, Vol. 53, No. 4, pp. 50-58, 2010.
[48] David Bermbach, Markus Klems, Stefan Tai and Michael Menzel, "MetaStorage: A Federated Cloud Storage System to Manage Consistency-Latency Tradeoffs", In Proceedings of the 4th IEEE International Conference on Cloud Computing, 2011.