IJCTA, 9(21), 2016, pp. 233-241

A Scalable Network Architecture using Replication in Virtual Environment

Nagamani H Shahapure* Dr. P Jayarekha**

***Abstract* :** Susceptibility to faults in cloud is a result of the increase in the number of applications deployed in cloud. This decreases the availability of resources to the user. As a result the scalability of the system is reduced. Fault tolerance and replication are fundamentals to maintain the availability and scalability of the system. This paper makes three contributions to address this issue of availability. Firstly architecture for a fault-tolerant framework in cloud is described. This framework automatically deploys replicas of virtual machines in data centers in a way that optimizes resources while assuring availability and responsiveness. A technique of virtualization and fault tolerance is proposed for availability. In this method the concept of heartbeat in a network is considered. Data is replicated across multiple servers. Whenever a server fails the heartbeat senses this. The control is transferred to another server which has the replicated data. The service is made available to the client through this replicated data. This technique provides scalability by using the concept of replication and availability.

Keywords : Availability, Scalability, Fault Tolerance, Replication, Heartbeat

1. INTRODUCTION

Cloud computing is a distributed computing paradigm that offers cost effective and scalable services to the users. It also provides a fault tolerant computing utility with globally uniform and hardware transparent user interfaces [1]. One of the important issues for a computing data center is the ability to provide a high availability service for all the applications running on it. This means that all the services should be available 24 hours a day and 7 days per week. This is irrelevant of the life span of the services. The services may run for a short time period or long period of time [2]. The data integrity of these services should be maintained [3]. This preserves the availability and increases the scalability. However the scaling up of data increases the complexity to achieve availability [4]. Data flows through the network from one location to another while using the services provided by the cloud. The data is prone to different types of errors. Outcome of these errors is failure of the system. The failure results in the halt of the services of the system. Maintaining copies of data in the network in a secured way is a crucial issue. Fault tolerance and virtualization techniques are available that replicates data at different location to tolerate data losses and ensures continued service [18].

Replication is a key mechanism to achieve scalability, availability and fault-tolerance [16] [17]. It is used to create and maintain copies of data at different locations [1] [13] [15]. Events occur that results in affecting a primary location where the data resides. Data cannot be retrieved from the primary machine. In such instances data is recovered from the secondary location. Endowing of continuous service is called as fault tolerance. Through this technique the data is recovered whenever there is information loss or failure. Fault can be tolerated and availability

* Department of Information Science and Engineering JSS Academy of Technical Education, Bangalore, Karnataka, India Email-nagamani1326@gmail.com

** Department of Information Science and Engineering BMS College of Engineering, Bangalore, Karnataka, India Email-jayarekha.ise@bmsce.ac.in

can be increased [1] [5]. This causes performance overhead as it takes time to recover data from other sites and restart the service again. We present the following contributions. In this literature review on fault tolerance, replication and placement of replicas in distributed system is introduced. Proposed algorithm for efficient dynamic replication and the topology used is explained in section II. Section III presents our contributions, experiment and results. The last section recapitulates the paper and gives a short overview of future works.

2. BACKGROUND

Fedrica [2] in his paper has mentioned about exploiting virtualization. This technique automatically reinstalls a host and provides a sort of host on-demand. This action is performed on a virtual machine only when a failure occurs. It is possible to achieve a redundancy system for all the services running on a data center by making use of virtualization.

Bakhta Merufel [6] mentions that in a distributed environment it is necessary to ensure continuous availability of data and respond to users quickly. An algorithm is proposed where the geographical distribution of nodes is considered. A technique of replication and assignment of data is proposed. The objective is to minimize the number of replicas that ensure certain availability degree without degrading the performance of the system. However task replication and fault tolerance of system jobs is not considered.

Laipong Zhao et. al. [7] has proposed a fault tolerant scheduling algorithm MaxRe. Here the reliability analysis is incorporated into the active replication scheme and exploit a dynamic number of replicas for different tasks. Compared to FTSA the reliability achieved is it uses 70% fewer resources. It does not however reduce the resource usage while running the reliability and deadline requirement.

YueGao et. al. in their paper [8] propose a scheduling algorithm that includes a static scheduling phase that operates on task graph based workload inputs prior to execution and a lightweight dynamic scheduler that migrates tasks during execution in case of excessive executions. The cloud service provider achieves high error coverage and fault tolerance using this method.

Asaf Sidon [9] proposed an algorithm called minicopyset. It is a simple and scalable replication technique. This method improves data durability and at the same time retains the benefits of randomized load balancing. This maintains the parallelization benefits of uniformly distributing chunks across the entire cluster. With this technique a system can achieve the same fault tolerance with 3 replicas that can be achieved with 5 replicas in random replication.

Roderigo N Calheiros [10] have implemented an algorithm that uses the idle time of provisioned resources and budget surplus to replicate tasks to moderate the effects of performance dissimilarity of resources on soft deadlines of workflow. This is also done to accurate the delays caused by underestimation of tasks execution time or fluctuations in the delivered performance of leased public cloud resource. Replication of tasks across multiple clouds is a drawback of this algorithm.

Frezewd Lemma et. al. [11] have put forth a dynamic replica placement strategy. A minimum repair time is taken during a complete loss of data center in large scale distribution storage systems. This is done besides maintaining the availability and durability of the system. The main focus is on minimizing repair times. This brings back the system to its most desirable state with high availability, durability, minimum network latency and load balancing.

Arindham Das and Ajanta De Sarkar [14] state that fault tolerance is the ability to preserve the delivery of expected services despite the presence of fault caused errors within the system itself. Breakdown and malfunctioning is avoided during the occurrence of faults. A fault tolerant service detects errors and recovers them automatically. Thus the system continues to deliver uninterrupted and acceptable services.

Rajalakshmi et. al [19] proposed a technique that provides incremental scalability and robustness to dynamic data in cloud. Dynamic data replication is used to improve data access in cloud. It concentrates on optimal replica selection and placement algorithm to increase availability of data in the cloud.

Frezewd Lemma [20] proposed a placement algorithm for micro data centers or micro clouds. An efficient replication strategy for the distributed storage system is proposed. This mechanism minimizes the repair time. High availability and durability is achieved while maintaining minimum network latency and load balancing. The duration of a repair process depends on the amount of data to be replicated and the transfer rate at which lost data is replaced. This is limited by reducing the storage capacity of micro clouds and will not be artificially restricted by the replication strategy.

Curtmola et.al [21] proposed Multiple-Replica PDP (MR-PDP) scheme. Here the data owner can verify that several copies of a file are stored by a storage service provider. Discrete replicas are created by first encrypting the data and then masking it with randomness generated from a Pseudo-Random Function (PRF). The randomized data is then stored across multiple servers.

Ayad F. Barsoum et al. [22] proposed creation of distinct copies by appending replica number to the file blocks and encrypting it using an encryption scheme that has strong diffusion property, *e.g.*, AES. Their scheme supports dynamic data operations but during file updates, the copies in all the servers should be encrypted again and updated on the cloud. This scheme suits perfectly for static multiple replicas but proves costly in a dynamic scenario.

3. PROPOSED ALGORITHM

3.1. Problem Formulation

In this paper an algorithm for dynamic replication is proposed. It uses the technique of fault tolerance and virtualization. Figure 1 shows the overall architecture of a fault tolerant system using replication. It consists of the client coordinator which is connected to the replicas. The replicas are virtual machines. The data is replicated across these machines. They take care that the response to the client's requests are not stopped because of any failures of the server. If any server fails, the task of serving the clients is immediately switched over to the replicas. This becomes a fault tolerant system which uses the concept of virtualization and replication.

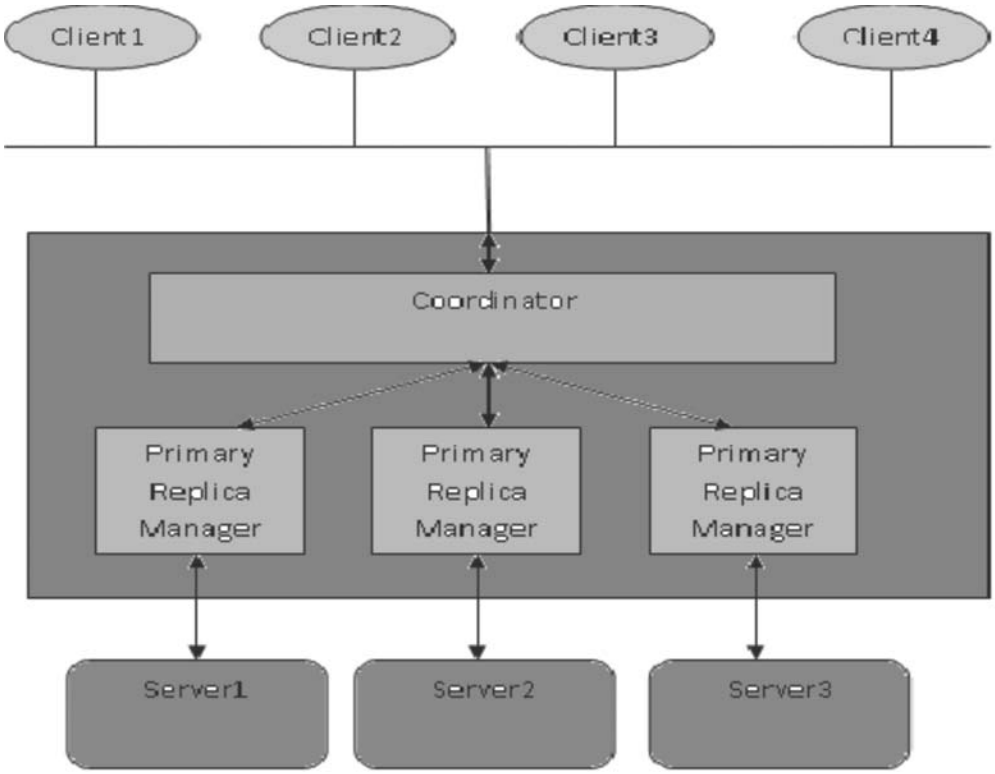

Fig. 1. Fault Tolerant System

A formulation is derived for the minimization of bandwidth of the virtual machines (VM). The variables used in the formulas are provided in table 1.

Table 1. Notations Used

Notations	*Meaning*
y_i^m	Required memory for the i^{th} client.
y_i^b	Required bandwidth for the i^{th} client.
φ_i^m	Portion of the VM memory assigned to the client.
φ_i^b	Portion of the VM bandwidth assigned to the client.
Mi	Memory usage of the i^{th} VM
$m_i^{'}$	Memory usage of the i^{th} VM replica.
M_j	Memory capacity of the j^{th} physical host.
b_i	Bandwidth of the i^{th} VM
$b_i^{'}$	Bandwidth of the i^{th} VM replica.
Bj	Bandwidth of the i^{th} physical host
Xi	Boolean integer to determine whether the i^{th} VM is on or off.
$x_i^{'}$	Boolean integer to determine whether the i^{th} VM replica is on or off.

The objective function is the minimization of the utilization of the bandwidth of the VM (Virtual Machine) and its replicas. Minimization of the bandwidth proves that the time consumed in serving the clients using the replicas is less compared to that of the physical server.

Minimize the bandwidth utilization :

$$\sum_{i=1}^{i=m} b_i b_i^{'}$$

Such that

$$\sum_1^m \varnothing_i^m = y_i^m \rightarrow \boxed{?}$$

$$\sum_1^m \varnothing_i^b = y_i^b \rightarrow \boxed{?}$$

$$\sum_{i=1}^{i=m} x_i = 1\,\forall_i \rightarrow \boxed{?}$$

$$\sum_{i=1}^{i=m} x_i^{'} = 1\,\forall_i \rightarrow \boxed{?}$$

$$\sum_{i=1}^m b_i = \sum_{i=1}^m b_i^{'} x_i^{'} = y_i^b \leq 1\,\forall_j \rightarrow \boxed{?}$$

$$\sum_{i=1}^m b_i x_i + \sum_{i=1}^m b_i^{'} x_i^{'} \leq y_i^b \forall \rightarrow \boxed{?}$$

$$\sum_{i=1}^m m_i x_i + \sum_{i=1}^m m_i^{'} x_i^{'} \leq M_j \rightarrow \boxed{?}$$

$$x_i \in (0,1),\ x_i^{'} \in \{0,1\},\ \varnothing_i^m \geq 0, \varnothing_i^b \geq 0 \rightarrow \boxed{?}$$

Equations 1 and 2 denote that portion of the memory and bandwidth allotted by the VM to the memory required by the client. 3 and 4 determine that every VM and its replica are deployed within a physical host. Equation 5 specifies the limit on the bandwidth assigned to the client by the VM and its replica. Equations 6 and 7 denote that the bandwidth and memory consumed by the VM and its replica is less than the bandwidth and memory of the physical host.

3.2. Algorithm

1. Start the Virtual Box

2. Create two nodes on the virtual box with replicated data.
   (*a*) Node1 with IP address 192.168.56.106. This is the primary node.
   (*b*) Node2 with IP address 192.168.56.105. This is the secondary node controlled by the heartbeat. It is used to provide service for users. It is disabled if this is the secondary node.
3. The client accesses the resource from the node which is located near to it.
4. If the node is overloaded then the request is transferred to the node which is under loaded.
5. If the node fails then the client request is served from the next nearest node.

4. EXPERIMENT AND RESULT

In our implementation of availability Oracle Virtual Box and heartbeat has been used. The replication technique used is a passive replication technique. Virtual Box is powerful *cross platform virtualization software* for x86-based systems. The heartbeat component of the Linux-HA (High Availability) is a simple program. Heartbeat is considered for checking the functioning of a node. It is designed to run continuously without memory leaks or bugs. It is easy to understand, easy to debug and extremely robust. The advantage of using heartbeat is it consumes less bandwidth. It is compared with other high availability techniques like Mysql Replication and Drbd (Distributed Replicated Block device).

To test for HA and scalability we have used the virtual box. We created three virtual machines on the virtual box. Each virtual machine has replicated data in it. Each VM senses the presence of other VM through the heartbeat. When a client requests for a resource the nearest VM serves the request. If the VM fails then the heartbeat of the next VM senses it. It takes over and serves the client. The replication used is passive replication. Two nodes are used. One is the primary node and the second one is the secondary node. Figure 2 shows the architecture of the proposed system.

Steps for configuring the nodes for checking the heartbeats:

1. Once a node is initialized, broadcast the IP address of the node to the port number 48514.
2. Listen to the broadcasted IP address. If the IP address does not belong to that server broadcast back the node's IP address to the node.

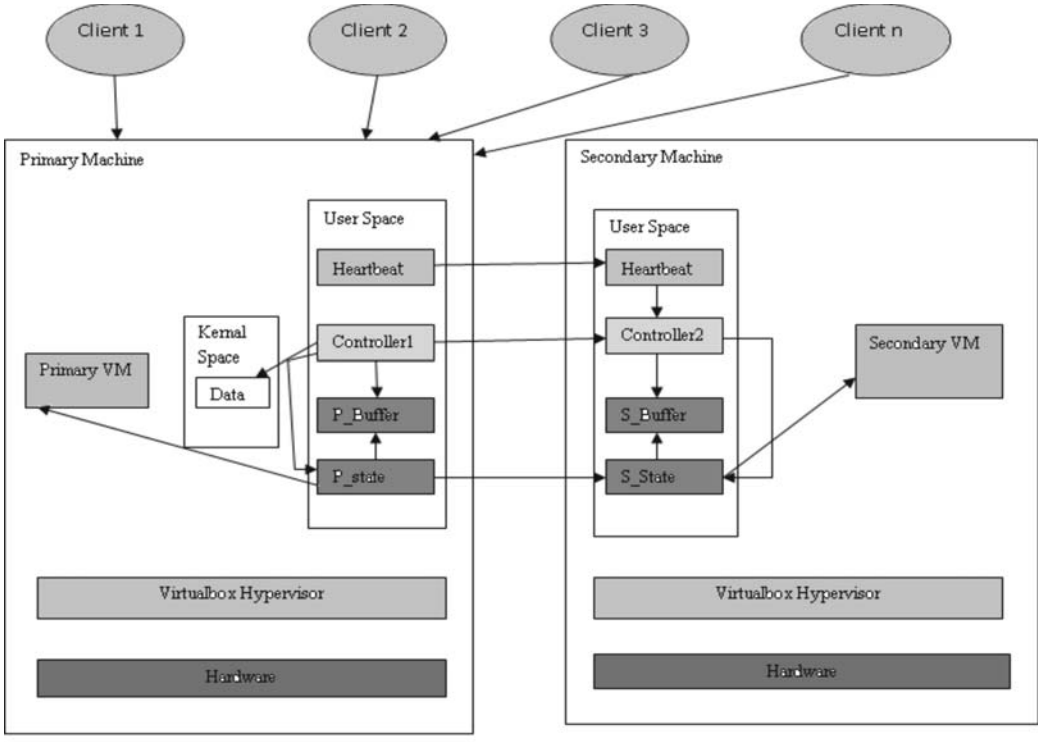

Fig. 2. Architecture of the Proposed System

3. Keep polling port 32000 for any broadcast to get paired.
4. Listen IP will get registered in the table.
5. Paired IP also gets registered in the table.
6. Registered and Paired IP will be used by the heartbeat to check node is alive.
7. Only once IP will be registered, check the table before updating the any new registration.

4.1. Bandwidth Usage

The formula for computing bandwidth used by heartbeat for N nodes on an un-switched network is as follows:

$$B_{hb} = S_{hb} * 8 \text{ bits} * R_{hb} * N$$

B_{hb} Bandwidth consumed in bits/sec

S_{hb} is the size of a heartbeat's packet in bytes is 150

R_{hb} is the rate of heartbeat per cluster node

A common heartbeat rate is 1 packet/sec; heartbeat packets average around 150 bytes. If a cluster has 1000 nodes with these other characteristics, then the B_{hb} for such a system is 1.2 × 106 bits/sec.

$$B_h = \text{Packet Size} * \text{Bit rate}$$

$$\text{Packet Size} = \text{Number of Bytes/Sample} \div N$$

$$(150*8)/1000 = 1.2 \text{ bits}$$

$$B_{hb} = 1.2*106\text{bps}$$

This is approximately 1.2% of the bandwidth available when other techniques like Drbd is used.

4.2. Procedure

Start the virtual box. Create the nodes in the virtual box. Configure the heartbeat in both the nodes. The two nodes are named as VMPS (primary node) and Node1 (secondary node). Run the VMPS. The IP address of this node is 192.168.56.106. It acts as the primary node. Run the secondary node Node1. This node has the replicated data. The IP address of Node2 192.168.56.105. Both have the same data. The nodes are configured for heartbeat. Run the client. The client is the web browser. The url accessed is 192.168.56.104/owncloud. This url searches the required page and serves the client. The data is fetched from the primary node. If this node VMPS fails or is overloaded then the request is transferres to the secondary node Node2. The failure of the node is sensed through the heartbeat. The switching of the transfer of data from one node to another consumes very little amount of time. The replication is dynamic. Figure 3 shows the details.

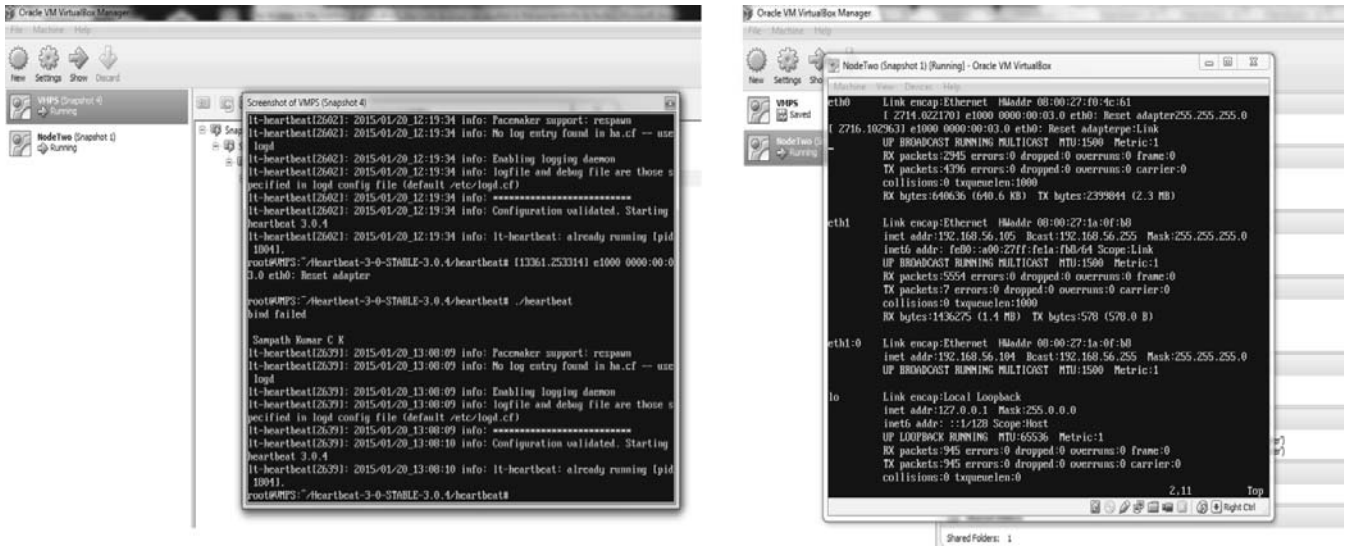

Fig. 3. (*a*) Primary Node (*b*) Secondary Node

(*c*) Switching from Primary to Secondary on the failure of Primary Node

The results of the experiment carried out using the three different techniques is shown in figure 4. The switching time from a failed node to working node is 3 seconds when heartbeat is used. This shows that availability is achieved better when heartbeat is used when compared with other high availability cluster techniques like Mysql Replication and Drbd (Distributed Replicated Block Device).

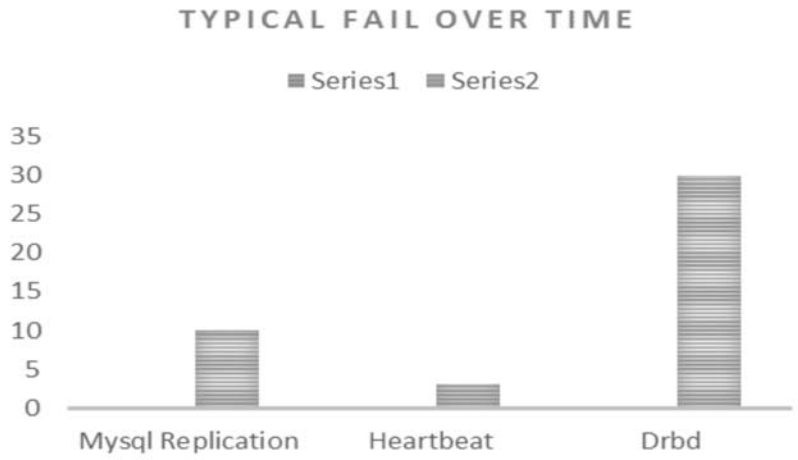

Fig. 4. Comparison of different techniques for switching over failed nodes

Table 2 shows the comparison of the different techniques for high availability with respect to scalability, failover time, availability.

Table 2. Comparison of Heartbeat with other Technology

Requirements	***Heartbeat***	***Mysql Replication***	***Drbd***
Availability	Yes	Yes	Yes
Automated IP Failover	Yes	No	Yes
Typical Failover Time	<3 secs	<10 secs	<30 secs
Automated Resynchronization of Data	No	No	Yes
Scalability	Yes	Yes	No
Geographic Redundancy	Yes	Yes	No
Support for Built in Load Balancing	No	No	No

5. CONCLUSION

A novel approach to maintain availability of cloud services by using replicas has been presented. This paper proves that the technique of replication using virtualization increases the availability. Availability improves the scalability. Heartbeat is the concept used for monitoring the working of the nodes. Once the heartbeat senses the failure or overloading of a node, then the task from that node is immediately switched to the replica. The advantage of using dynamic and passive replication is the bandwidth utilization is reduced by 1.2% in the virtual servers. It can recover from a failure with < 3 seconds of the downtime of the primary machine.

Further goal of this paper is to support replication beyond fault tolerant. As a future work different methods should be incorporated for better cooperation and reliability between different VM copies. This can be done by making better utilization of the virtualization technology.

6. REFERENCES

1. Dhananjaya Gupt, Mrs.Anju Bala, "Autonomic Data Replication in Cloud Environment", International Journal of Electronics and Computer Science Engineering, ISSN:2277-1956 //V2N2-459-464.
2. Federico Calzolari, "High Availability Using Virtualization", Ph.D Thesis, University of Pisa.
3. Raghul Mukundan, Sanjay Madria and Mark Linderman, "Replicated Data Integrated Verification in Cloud", Journal of Distributed and Parallel Databases, Vol: 32, Issue: 4, Pages: 507-534, Dec 2014DOI: 10.1007/s10619-014-7151-0.
4. Nicolas Bonvin, Thanasis G. Papaioannou and Karl Aberer, "A Self-Organized, Fault-Tolerant and Scalable Replication Scheme for Cloud Storage", SoCC 10, Proceedings of the 1st ACM symposium on Cloud computing, Pages 205-216, ISBN: 978-1-4503-0036-0 doi:10.1145/1807128.1807162.
5. Sheida Dayyani and Mohammad Reza Khayyambashi, "A Comparative Study of Replication Techniques in Grid Computing Systems", International Journal of Computer Science and Information Security, Vol:11, No:9, September 2013.
6. Bakhta Meroufel and Ghalem Belalem, "Managing Data Replication and Placement Based on Availability", AASRI 2013 Conference on Parallel and Distributed Computing Systems, Elsevier, Pages: 147-155, 2212-6716 © 2013, doi: 10.1016/j.aasri.2013.10.071.
7. Laiping Zhao, Yizhi Ren, Yang Ziang and Kouichi Sakurai., "Fault-Tolerant Scheduling with Dynamic Number of Replicas in Heterogeneous Systems", 2010, 12th IEEE Conference on High Performance Computing and Communications, 978-0-7695-4214-0/10 $26.00 © 2010 IEEE, DOI 10.1109/HPCC.2010.72.
8. Yao Gao, Sandeep K. Gupta, Yanzhi Wang and Massoud Peddram, "An Energy-Aware Fault Tolerant Scheduling Framework for Soft Error Resilient Cloud Computing Systems", Design, Automation and Test in Europe Conference, IEEE 2014, Pages:1-6, INSPEC Accession No: 14253587, doi: 10.7873/DATE.2014.107.
9. Asaf Cidon, Ryan Stutsman, Stephen Rumble, Sachin Katti, John Ousterhout and Mendel Rosenblum, "MinCopysets: Derandomizing Replication In Cloud Storage", Stanford University.
10. Rodrigo N Calheiros and Rajkumar Buyya, "Meeting Deadlines of Scientific Workflows in Public Clouds with Tasks Replication", IEEE Transactions on Parallel and Distributed Systems, July 2014, Vol:25, Issue:7, Pages: 1787-1796, ISSN: 1045-9219, INSPEC Accession Number: 14396643, doi: 10.1109/TPDS.2013.238, IEEE Computer Society.
11. Frezewd Lemma, Johannes Schad and Christof Fetzer, "Dynamic Replication Technique for Micro-Clouds Based Distributed Storage System", Third International Conference on Cloud and Green Computing, IEEE, 2013, Pages: 48-53, INSPEC Accession Number: 13991489, doi: 10.1109/CGC.2013.16.
12. Shweta Jain, Ashok Verma and Rashween Kaur Saluja, "Fault Detection and Tolerant System (FDTS) for SaaS Layer in Cloud Computing", International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 3 Issue 8 August, 2014 Page No. 7938-7942.
13. Adrian Coles and Bica Mihai, "An Adaptive Virtual Machine Replication Algorithm for Highly-Available Services", Proceedings of the Federated Conference on Computer Science and Information Systems pp. 941–948, ISBN 978-83-60810-22-4, 978-83-60810-22-4/$25.00@ 2011 IEEE.

14. Arindam Das and Ajanta De Sarkar, "On Fault Tolerance Of Resources In Computational Grids", International Journal of Grid Computing & Applications (IJGCA) Vol.3, No.3, September 2012, DOI: 10.5121/ijgca.2012.3301.

15. Sushant Goel and Rajkumar Buyya, "Data Replication Strategies In Wide Area Distributed Systems", Enterprise Service Computing: From Concept to Deployment, doi: 10.4018/978-1-59904-180-3ch009, 2007.

16. Xavier Defago, Andre Schiper, Nicole Sergent, "Semi-Passive Replication", 1998, IEEEComputer Society Press, Appeared in Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems.

17. UmarFarooq Minhas "Scalable and Highly Available Database Systems in the Cloud", Ph.D Thesis in Computer Science, Waterloo, Ontario, Canada, 2013.

18. Andrew J. Younge, Robert Henschel, James T. Brown, Judy Qiu and Geoffrey C. Fox, " Analysis of Virtualization Technologies for High Performance Computing Environments", In: Proceedings of The Fourth IEEE International Conference on Cloud Computing (CLOUD 2011), Washington , Washington DC, USA, July 4-9 (2011); technical Report (February 15, 2011), updated (April 2011).

19. A. Rajalakshmi D. Vijayakumar and K. G. Srinivasagan, "An Improved Dynamic Data Replica Selection and Placement in Hybrid Cloud", International Journal of Innovative Research in Science Engineering and Technology, Vol.3. Special Issue 3. ICIET'14, 2014.

20. Frezewd Lemma, Johannes Schad and Christof Fetzer, "Dynamic Replication Technique for Micro Clouds Based Distributed Storage System", Funded research as part of LEADS project supported by the European Commission under the Seventh Framework Program (FP7) with gant agreement number: 318809.

21. R. Curtmola, O. Khan, R. Burns and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession", 28th IEEE ICDCS, pp.411_420, 2008.

22. A.F.Barsoum and M.A.Hasan, "On Verifying Dynamic Multiple Data Copies over Cloud Servers", 2011.