International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 06 | June -2020 ISSN: 2582-3930

Data Replication in Cloud Computing

Apeksha Ganiger[1], Dr. Usha J[2]

Master of Computer Applications, RV College of Engineering , Bengaluru

Abstract -Cloud computing is a system that generates and uses large quantities of information every day. It helps the cloud hold tons of users. The market for those sources is growing. Multi-cloud surroundings are used to satisfy these requests. If a lot of vendors work together, resource quality may become better. Replication of statistics over more than one mesh locations has emerged as a systematized or a productive approach to harvest sufficient overall output in traffic stabilize phrases, reply period, and accessibility. Truth Replication is a satisfying method of gaining trustworthy and increasing overall efficiency in a shared structure. Cloud computing's growing acceptance is an opportunity for conventional fact-refinement structure. Replication has expanded the worth of their structured and uninterrupted service, even when there are faulty components. Fault tolerance is a major challenge in ensuring critical services are accessible and secure, as well as program execution. To limit system failure and utility execution, faults need to be expected and Handled proactively.

***Key Words*:**Cloud, Replication, Fault Tolerance, Performance Enhancement, Data Availability.

1.INTRODUCTION

For users who use the cloud to store the facts, the main challenge is to protect the information and regain where appropriate. Some crack to the server does not result in loss of data. All cloud services include play, voice and video conferencing, online workplace, storage, backup, and social networking. Such applications' effectiveness depends primarily on the accessibility of excessive effective communications services and network regulation [1]. Information replication is a widely utilized method to boom the accessibility of facts. It calls for an excessive information rate direction of the bandwidth. Cloud duplicates the records, strategically keep facts on various servers pinpointed at different places [24]. Replication guarantee stability; boosts accessibility and dependability through making multiple copies on distinctive storage devices and geographic locations of the same truth. Replication and quality take a key part in the acceptance to faults [16]. The data is delivered over the web. This must be made available to those programs which need to apply it. No need to degrade the output now. The records get admission to speed need to be expanded, maintaining the traffic stability within the structure. Expandability and accessibility are the two major elements to upgrade the execution of the cloud. Generating copy's are likewise the critical strategies to obtain the best performance. Generating the duplicates or replicas as well lessen get entry to intermission and high frequency utilization [17].

2.REPLICATION IN CLOUD

Replication produces several replicas of an established organization [2]. Replication can increase the resource obtainability and as well gives stability and accuracy by generating alike information many copies of particular websites. Replication as well allow low value entry, mutual usage of bandwidth and delay time by statistics replication. If there occurs damage of the system, the cost of duplication is to have simple, seamless permission to use the resources. Replication can be applied through a computer network in order to be able to position storage devices in actual different provisions. Clients control close by copies, and enlarges productivity if record transmission is not preserved. Saving the data at different locations has advantages. If a server with the information needed fails, a device will use the replicated information to perform. This definition preserves transparency. The documents are kept at many locations. Through the closest supply through which the appeal emerged, the required statistics are obtained. That will increase the gadget's overall efficiency. Replication benefits don't come without the overheads to build, maintain, and repair replicas. Replication will significantly boost performance [3].

2.1 Role of Replication in Cloud[19]

1]Uptime and resiliency

Organizations that can be continually pushed to staycompetitive are still searching for ways to be creative and retain a high degree of uptime. If it's miles of redundant equipment or a personal hot-web, website, it's a hard job to keep an infrastructure up and running 99.99 percent of the time (insert extra 9's here). Cloud computing clearly will help here. An IT system can reproduce the website-to-website surroundings correctly by using either private or public cloud technologies. It can be done via a cloud provider to a specific web page or one hosted on licensed websites. With a properly deliberate deployment and a great infrastructure, agencies can effectively load their IT environment between more than one active, totally cloud-based websites. And if one website were to step down – users would be transparently matched to the nearest or most accessible center of facts afterwards[25].

2]Remote backup and storage[21]

Structures for storage have become more efficient and can now do offsite, cloud backup and replication. The use of cloud replication may be an option for those companies trying to take offsite of their surroundings. The engineers are able to completely backup and even restore from the cloud provider by providing a dedicated hyperlink to a web-based totally information center. In addition, certain businesses are bound

by such rules for the preservation of information. This is also where cloud garage can help. The documents are retrievable (now, however, not for DR purposes) and can be checked as appropriate. Using cloud-based fully backup and replication provides a robust system capable of maximizing recoverability and continuity of business. The modern-day market for primarily cloud-based backup and garage is developing and more firms offer competitive pricing. Organizations can undertake a flexible boom plan capable of scaling statistical demands on the IT infrastructure.

3]Remote workplaces

Remote workplaces can benefit from a non-public cloud environment where middle facts deliver information, workloads or even computer systems to remote users and branch workplaces. In the business information center, most of the horsepower is achieved by getting just a few coordinating variables at the degree of the branch workplace. Some computers or main servers should also stay as nice as feasible as a WAN optimizer in the remote office [23]. This type of effective communication technique reduces the amount of critical infrastructure additives in remote workplaces. Additionally, the use of cloud replication and fewer branch additives allows administrators to monitor sources and very granular control them. This could have exceptionally good cost savings for a company if properly applied and deliberated out.

4]Building a "business-in-a-box"

The concept behind cloud computing helps many managers to simplify the launch of a new business division when developing a "box-outdoor" organization. By centralizing the entire system within a cloud environment, IT needs to deploy some end-factor additives to provide the most efficient way of accessing the cloud-hosted platform. This solution will contain workloads, databases, desktops, applications and everything else needed to manage an organization's day-to-day business enterprise activity. This will cut the standard business launch plan in time and dramatically expedite it. We are raising the amount of hardware we need and increasing agility by using the cloud to represent commercial business processes. The splendor of this form of state of affairs is that the entire network can be housed in a non-public, public, or even hybrid cloud environment.

2. 2 Benefits of Replication in Cloud

2.2.1.Data Availability[20]

The product availability is determined through the amount of time the company is in operation by using the customers under both usual and odd situations. The high availability derives from the fact that its customers require constant access to the service [4]. The lack of availability of other providers has a negative effect on their customers [1]. It is valid for financial institutions, telecommunications firms, software from the army or hospitals. Cloud infrastructures provide more than 99.9 percent of availability; consequently, degradation of performance is more serious than resource disasters in such environments. There is an apparent increase in requirement for uninterrupted availability of HPC structures. This is a first-rate way of computing capabilities, in which scientific programs want lots of time (weeks and months) with uninterruption on the rapid available HPC machines. In the event of failures these excessive quit computing (HEC) systems should be able to operate in such a way that their capacity is not fastidiously diminished. In project-critical uses, like in the telecommunications industries, high-availability (HA) computing has long-term played a crucial role. A large array of replicas is saved through two or more servers to reach more availability. If a server unavailability exceeds a specified time, then automatically starts the replication process. That guarantees continuous operation.

2.2.2Fault Tolerance[22]

The effort to build distributed fault-tolerant systems began a long time ago. A possible security vulnerabilities are a big problem inherent in distributed systems. Nevertheless, if a single node fails, the functionality of the entire network may be jeopardized [5]. But, the distributed design of these systems means that the gadget 's efficiency can develop. A tolerant fault machine is a mechanism that saves a computer gadget from losing within due to a surprising issue. It is the capacity to maintain the carriage of presumed deals, given the existence of errors resulting consequences in the program. It targets, avoiding failures within the availability of errors and these are found in a this system, and corrected. Fixed errors are put and cleared even the work proceeds to produce suitable offerings. It is related to all the methods required to allow a gadget to hold the software errors that remain in the system after its evolution [15].

This can be categorized as reactive tolerance of failures & constructive tolerance of failures [6].

1] Reactive Fault Tolerance [18]: It's a method that reduces the effect of faults on application work whilst the completely happens. Checkpoint / Restart is the method implemented. If a venture doesn't work, it's permitted to re-start instead of starting of the element fault. Another solution is to retry. The failed technique in this technique is re-tried in the particular tool. Re-submission of tasks is a different way. If an improper project is identified, it is re-submitted at runtime. Another method for reaching this is through the process called replication.

2] Constructive tolerance of fault: The idea of constructive tolerance of faults is to prevent fault recovery. Working additives alter the defective additives. One way to do this is by using a method known as rejuvenation of the software program. The system is conceived here for periodic reboots. The machine reboots in smooth shape. The approach used for constructive acceptance of blame is by self relieving. A program runs on different virtual mechanisms, multiple times. If one VM doesn't work the opposite takes over suddenly. Fault tolerance, durability and scalability proportional. All of these boundaries are essential to ensuring proper, vigorous working of the system. The quality is calculated in intermediate timing between failures and the interim repair time [7]. The high availability is discussed by server replication and storage [8].

2.2.3.Enhance performance

More load (example: user requests) can be accepted because there is a sharing of the workload among several processes. Additionally, with the help of replicating facts in the user direction, latency can be reduced. Unfortunately, if the records are read / write, benefits are reduced [26].

3.REPLICATION TECHNIQUES [5] [8]

1.Active Replication[14]

Active Replication is also known as state-machine replication introduced by Leslie Lamport. All servers manage each client request. Every nature of replicas is independent of one another. It is important to ensure that all replicas receive the requests in the same order. An atomic broadcast protocol should be used to have all servers receive the same sequence of operations. A protocol for atomic broadcasting guarantees that either all servers get hold of a message or none, plus that all servers receive messages in the same order. The major drawback for this method is that majority about the actual international servers are non-derivable during testing. Though energetic duplication is the preferred desire while proceeding with concurrent structures requiring a brief response even below, the presence of faults or with fault-care structures [28].The downside of this replication: (1) the filtering consistency means a huge use of help (2) the appeals must be managed in a derivable way. Derivable because the end outcome of a performance hangs on most handily on the starting position of a duplicate and the series of processes all have so far carried out.

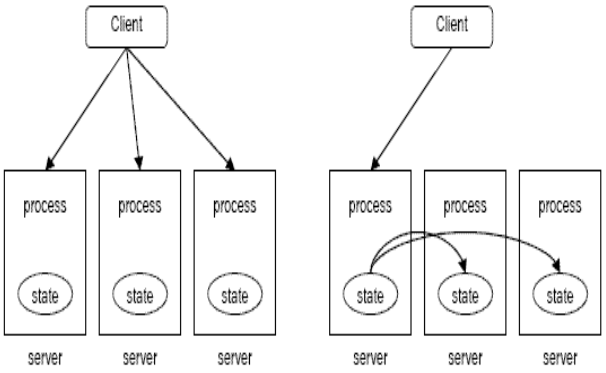

Fig -1: Active and Passive Replication[9]

Steps in active replication functioning[9]:

1.Request: The front end adds a one of a kind identifier to, the solicitation and broadcasts it to the imitation directors, network, utilizing a strictly organized, robust primitive multi cast. At worst, the front end is presumed to break by failure. The next request is not given unless front end has obtained a reply.

2.Coordination: The people group correspondence framework will transmit the solicitation in the equivalent (all out) request to any correct Replication manager.

3.Execution: The appeal is executed by each Replica Manager. Because those exist as state machines and because applications are supplied in the similar complete order, the request is handled identically by replica managers. The response contains the unique identifier of the client's request.

4.Agreement: There is no need for a process of agreement due to the semantics of multi cast delivery.

5.Response: Every Replication manager posts response to the front end. The count of response collected by the front end relies on the supposition of failure and working method for multi cast. For instance, if the objective is about the accept at most failures due to crash and the broadcast follows standard unity and structuring possessions, after which the front end transfers the initial answer to the customer and throw away the remainder.

2.Passive Replication[9]

Here is just a single server, for example essential server, that forms customer requests and keeps the reinforcement by refreshing the state on different servers and answering the message back to the customer. At whatever point essential crashes, it takes its place on one of the backup servers. Passive replication might also be utilized as the primary backup for non-derivable. Passive replication plays a primary role receiving the requests [9].

Non-deterministic [9]: The main back-up is often called passive replication. In which it plays a primary function that gets the customers requests and returns to them. This method of replication is better than active replication because it needs slighter operating ability and builds no statement about the determinism about an appeal being processed.

I Front ends exchange with one, primary replica manager, within the passive replication model. The essential replica manager answers appeals, then sends updates to many managers of secondary replicas. A backup (secondary) replica manager can take his place in case the first fails.

Working of passive replication:

1. Request: The front end sends the appeal for the primary replication manager, which contains a completely unique identifier.

2. Coordination: The number one draw every appeal spontaneously, within the sequence the way that occurred. It verifies the individuals, if the request has so far been reached and in that case it actually again forwards the reply.

3.Execution: The essential performs the appeal then the reply is stored.

4.Agreement: In the event that the request is an upgrade, the main must send the upgrade, answer and one of a kind individual for all reinforcements. The reinforcements express an affirmation.

5.Response: The number one answer to the front end, palming again the reply to the client.

2.2 Active Replication VS Passive Replication

Active replication requires to be deterministic about the operations on the replicated item. Determinism how the final results about the functioning depends solely on the item 's first status together with the order of functions conducted through the item (record). In the event of functions at the duplicated item are derivable, the replicated item's distributed position remains steady and every reply posted to the customer are similar. Consequently, from any replica the user might practically look ahead to the first answer. A fascinating property of successful replication is the reality means a failure no longer booms the intermission encountered by a user [30].

Passive replication no longer wastes more resources through redundant processing as opposed to active replication, and allows non-deterministic operations. A number one crash, however, could dramatically boom the latency of an invocation as well. In addition, passive replication calls for additional number one application support, to update the status of the alternate copies.

Both replication approaches have a trade-off. Active replication has a few reliable and typically a quicker reaction period, compared to passive, which calls for less support for implementation. On the contrary, passive replication is highly scalable, as it may not expect servers to be derivable any more. Because active and passive replication complements each other, accepting both methods is a prime gain which allows the technician to select the method that is suitably adjusted to the complications [29].

3.Quoram Replication

In the Quoram replication, Each slave(client) reads from r masters(servers) and writes to w masters(servers).If r+w>n, then the intersection of each pair of read/write sets is non-empty and each study will see at least one replica of the latest written price.The r=1 and w=n imply complete replication (write-all, study-one): Unwanted when servers may be inaccessible due to the facts written are not guaranteed to be complete. The best performance (throughput / availability) when 1 < r < w < n is exceptional due to the facts read as written in most applications. The generalization: r, w spread through customers but non-empty intersection of all read / write sets is retained.

Fig -2: Serial Consistency(R1, R2 both read value written by W3)

Imagine, for example, a community of 3 réplicas. Then we have the possibilities which follow:

R=3, W=1. It increases overall output for writing at the cost of reading, which may be a poor move because usually reading is not more special than writing. Moreover, this desire for quorums is bad because a writing could manifest in a single duplicate, which then fails. If that replica were to lose its place, it would lose the final write performance. So we'd really like to have W>1 generally.

R=1 and with W=3. It fits very well for readings which are generally suitable on the grounds that it reads. But because of the fact that if one of the replicas is down or inaccessible, a writing can not complete until that duplicate recovers, it's far undesirable for writes.

R=2 and with W=2. This desire is a strong option compared to the preference for R=1 and W=3, because it reduces the cost of reading in exchange for fair written availability.

3.1Protocol at the replicas

The replica returns the cost with its version quantity in response to a read request for a few items (or simply the model variety if that is what the clients want).

If the new version wide variety is larger than what the duplicate has stored in response to a write request, the replica overwrites the new edition range and new condition. Otherwise, it does not delete but will just continue what it has. In either case an acknowledgment is returned.

3.2Performance Optimizations

It can be big data items, e.g. During the quorum protocol service, files and several copies are transmitted by extremely inappropriate use of the bandwidth

1.Hashing

i)Every server stores hash of data and information

ii)Replies hash values and meta (not records) statistics at some point in the study operations

iii)Voting entirely based on hash data

iv) When correct hash has been calculated, query a one master for record item and measure its hash to test the validity of the facts

2.Caching

i)Store the items in caches for purchasers

ii)Perform hashing arrangement for a asses

iii)If item is in the cache, calculate the cached item hash and no need to query an object server now if the stocked copy is the same as the date.

3.3Recovery

This protocol has the assets that operations continue to function properly when a replication fails (or is inaccessible because of a network problem). If the quorum of replicas required is available, complete operations; in any other case, their execution could be delayed until the failure or inaccessibility is over.

Some care is needed when a reproduction recovers: in order for the protocol to function properly, it must be the case that a replica only responds to requests after it is aware of all the adjustments it recounted earlier than the failure.

One way to fulfill this requirement is to write the completely newer state rather than submitting the acknowledgment for a modified object to disk. Then on recovery the duplicate can immediately keep offering provider, as long as its disk has not been trashed now.

If there is a problem with the disk, however, or if the replication sends acknowledgments earlier than writing to the disk, then the duplicate must get all the changes it may have lost. It can try this by researching all of the opposite replica devices. This examination does not require a write-back, because the write-back will be done by next exam accomplished by way of a client.

The purpose of the replica is to improve its status earlier than responding to requests from patrons is that otherwise it would matter in a quorum, however the information it sends far may be stale.

3. 4 Drawbacks of Quoram Replication

Unfortunately, QC has some features that are not so outstanding now, too. Except in simple cases, a front-end client has to permission for any number of copies of every object of facts that it wants to read. If there exists duplication of the knowledge item on the company's website (i.e. the company is also a replica), the user front-stop will also search for other copies elsewhere.

So, it could build a quorum for reading. Customers read more pieces of information in many applications than they write. Such packages may not now be performing well with QC. User may face this problem by recommending that one read each x quorum contains a single duplicate of x. But then one

compose quorum can must be for x, one that involves all duplicates of x. This would lead to write-all approach, which was unsatisfactory to the discovery.

An other issue with QC is that it wants a wide variety of duplicates to abide at any amount of website faults. For example, suppose all majority sets are quorums. Then QC would like 3 duplicates to undergo one failure, 5 copies tolerate faults, and so on. Copies in particular are no useful at all. QC can't even handle one mistake of copies.

An another problem with QC is that everyone copies of every statistics object ought to be remembered in advance. A known replication of x will heal, but a late duplicate of x can't be generated because it can change the concept of x's quorums. In principle, while the replicated gadget is running, one can alternate the value of the sites (and concept of quorums) but this needs important synchronization.

4.Lazy Replication[10]

Often known as optimistic replication is a method as replication, in which replicas are allowed to diverge [27]. The lazy technique is believed so as the server responds request of the user once the data update is done domestically. Once a scan or reading process is given to one among other copies by the client before the restore propagation the old information will be visible.

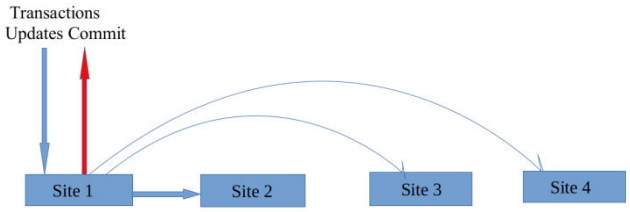

Fig -3: Lazy Replication first executes update transaction on one copy

Lazy replication – underlying idea is that if update has been successful on one node, it will eventually be successful on all other nodes as well.

Bottom line analysis

Amount of simultaneous exchanges starting at a hub:

Transactions = Actions x Action Time = TPS

N nodes = > N X # transactions come from every second.

Each user transaction in Lazy system generates N-1 lazyreplica updates = > N nodes generate N

Each time there are more transactions, then N^2 again.

5.Chain Replication[11]

Chain replication is another way to deal with the management of failure-end server clusters. The methodology is designed to help enormous scope storerooms that display high limit and accessibility without trading off away from of consistency. Besides sketching out the chain replication conventions themselves, reproduction examines are exploring a model implementation's productivity traits. Efficiency, accessibility, and many article position techniques (counting plans concentrated on the routing of hash table distributions). This methodology accepts that in an arrangement/chain imitations are organized. A chain comprises of a lot of hubs with a replacement and an antecedent in every hub aside from the main (head) and last (tail) hubs of the chain.

Storage carrier clients issue queries, requests and update activities. In spite of the fact that it is conceivable to ensure that any appeal that enters the repository system is ensured that end to end row will be executed [12]. Customers are higher off if the capacity, administration certainly creates a reaction for each solicitation it gets and finishes, since this takes into consideration lost demands and answers to be treated to: If an excessive amount of time has slipped by without getting a reaction, the server retrieves the request.

A customer appeal which is missed earlier than the storage carrier is identical from one which is disregarded through the garage service to that customer. The approach whereby consumers should not now be discovered in a brand new failure mode when a garage server for the period of which customer requests are ignored is known as brief outages. Although, likely sustainable customer function may rely on outlawing the occurrence and period of temporary interruption.

With chain replication, a way shorter than the time needed to dispose of a defective host or feature a brand-new host is the duration of each brief outage. So the processing of client requests proceeds with minimal disruption in the face of failure, recovery and different reconfiguration. Most other replica-control protocols either block a few operations or the consistency of sacrifices ensures after disasters and throughout reconfiguration.

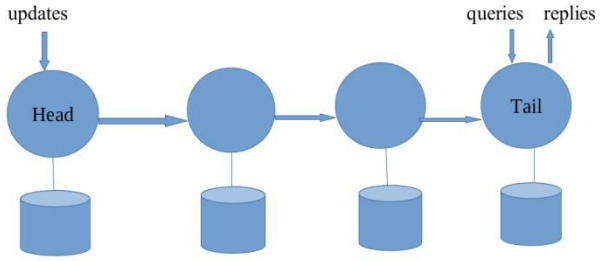

Fig -4: A Chain

Figure 4 determines the objID status in terms of variables: the list Hist objID of changes made on objID, and a set Pending objID of unprocessed requests.

The figure then lists feasible transformations in the State. Transition T1 states that PendingobjID is sending an arriving customer order. Use transition T2 states that a few pending requests are ignored-this change is hopefully no longer taken too often. Transition T3 provides a high-degree view of the processing of requests: the request r is first removed from PendingobjID; query, then causes a suitable answer to be generated while update also adds r (referred to as ·) to HistobjID.

5.1 Chain Replication Protocol[11]

Generation of Responses. For each question, the answer is generated and sent through the tail.

Processing query. Every query request is guided to the tail of the chain and the use of objID replica saved at the tail is processed atomically there.

Processing update. Each request for replacement is addressed to the chain header. The request is processed atomically using

objID replacement at the head, then state changes are transmitted along a secure FIFO connection to the next chain information (where it is treated and transmitted), and so on until the request is treated using the tail.

4. CONCLUSIONS

The Cloud is a type of disbursed computing, which is earning popularity. Like most of the services installed in the Cloud, there has been an unprecedented rise in the amount of users utilizing this cloud. Scalability and availability of such applications are needed. One approach to obtaining these parameters is the replication of the offerings. One of the crucial concepts is to apply replication and fault tolerance of availability. The different replication techniques play a different role and each of them has different characteristics, limits and delimits. It is hard to say that one is better than another because each of them has a special role in replication.This can be applied to heartbeat. This concept of heartbeat can be utilized in cluster computing. It's based primarily on a high-accessibility active-passive results. Every provider in this high-availability requires minimum of two equivalent servers, a number one host running the carrier, the other secondary host running, capable of improving the uses in within a second. Heartbeat machine is utilized to reveal the node fitness. It's capable of detecting problems or disasters of single node & routinely switching workloads or packages in the same cluster to another energetic node. These systems usually have redundant hardware and software which, despite the failures, keeps the programs available.

REFERENCES

1. Nagamani H Shahapure and P Jayarekha, "Replication: A Technique for Scalability in Cloud Computing",International Journal of Computer Applications (0975 – 8887) Volume 122 – No.5, July 2015
2. Arindam Das and Ajanta De Sarkar, "On Fault Tolerance of Resources in Computational Grids", International Journal of Grid Computing and Applications, Vol.3, No.3,Sept. 2012, DOI: 10.5121/ijgca.2012.3301
3. Sushant Goel and Rajkumar Buyya, "Data Replication Strategies in Wide Area Distributed Systems", Grid Computing and Distributed Systems Lab, University of Melbourne, Australia
4. 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 c 2011 IEEE
5. Xavier D´efago, Andr´e Schiper and Nicole Sergent, "Semi-Passive Replication", IEEE Computer Society Press. Appeared in Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems
6. Thakur Kapil Singh, Godavarthi Tarakarama Ravi Teja and Padmavathi Srinivasa Pappala, "Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing", International Journal of Scientific and Research Publications, Vol.3, Issue 6, June 2013, ISSN 2250-3153
7. R. Jemina Priyadarsini and L. Arockiam, "Failure Management in Cloud: An Overview", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 10, October 2013, ISSN (Print): 2319-5940, ISSN (Online): 2278-1021
8. Sergio Filipe Garrau dos Santos Almeida, "Geo Replication in Large Scale Cloud Computing Applications", Dissertation submitted for Master Degree in Information Systems and Computer Engineering,November 2012
9. KaranjitSingh Tuwana, Geetha BK, Shreesha Bhaskar, Rakshita GS and P Beaulah Soundarabai, "Comparative Study on Active and Passive Replication", International Journal of Engineering Research & Technology (IJERT),Volume 4, Issue 21, ISSN: 2278-0181,ICRET - 2016 Conference Proceedings,Published by www.ijert.org
10. Esther Pacitti , Pascale Minet, Eric Simon " Replica consistency in Lazy Master Replicated Databases distributed and Parallel Databases", 9, 237267
11. Robbert van Renesse , Fred B. Schneider, "Chain Replication for Supporting High Throughput and Availability",FAST Search & Transfer ASA Tromsø, Norway and Department of Computer Science Cornell University Ithaca, New York 14853
12. J. Saltzer, D. Reed, and D. Clark. End-to-end arguments in system design. ACM Transactions on Computer Systems, 2(4):277–288
13. "Quorum Consensus Replication" Material de apoio complementar para a cadeira de Sistemas Distribuídos, DEI, IST, 2012/13
14. Andre Martin and Christof Fetzer,Andrey Brito, "Active replication at (almost) no cost",2011 30th IEEE International Symposium on Reliable Distributed Systems
15. Wenbing Zhao,P. M. Melliar-Smith and L. E. Moser, "Fault Tolerance Middleware for Cloud Computing",2010 IEEE 3rd International Conference on Cloud Computing
16. Wenhao LI , Yun Yang, (Corresponding author), Dong Yuan, "A Novel Cost-effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres ",2011 Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing,978-0-7695-4612-4/11 -2011
17. Dejene Boru1 , Dzmitry Kliazovich2 , Fabrizio Granelli3 , Pascal Bouvry2 , Albert Y. Zomaya4, "Models for Efficient Data Replication in Cloud Computing Data centers ",IEEE ICC 2015 - Communications QoS, Reliability and Modeling Symposium,978-1-4673-6432-4/15/ -2015
18. Das, P., & Khilar, P. M. (2013). "VFT: A virtualization and fault tolerance approach for cloud computing", 2013 IEEE CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES.doi:10.1109/cict. 2013.6558142
19. Tabet, K., Mokadem, R., Laouar, M. R., & Eom, S. (2017)."Data Replication in Cloud Systems",International Journal of Information Systems and Social Change Volume 8 • Issue 3 • July-September 2017,doi:10.4018/ijissc. 2017070102
20. Amjad, T., Sher, M., & Daud, A. (2012). A survey of dynamic replication strategiesfor improving data availability in data grids. Future Generation Computer Systems, 28(2), 337–349. doi:10.1016/j.future.2011.06.009
21. Milani, B. A., & Navimipour, N. J. (2017).A Systematic Literature Review of the Data Replication Techniques in the Cloud Environments. Big Data Research, 10, 1–7.doi:10.1016/j.bdr.2017.06.003
22. Youssef M.Essa, "A Survey of Cloud Computing Fault Tolerance: Techniques and Implementation",International Journal of Computer Applications-2016,DOI:10.5120/ijca2016909055
23. Mo¨ıse W. Convolbo, Jerry Chou, Shihyu Lu† and Yeh Ching Chung, "DRASH: A Data Replication-Aware Scheduler in Geo-distributed Data Centers",IEEE 8th International Conference on Cloud Computing Technology and Science-2016,2330-2186,DOI 10.1109/ CloudCom. 2016.53
24. Faraz Fatemi Moghaddam,Mohammad Ahmadi,Samira Sarvari,Mohammad Eslami,Ali Golkar,"Cloud Computing Challenges and Opportunities: A Survey",1st International Conference on Telematics and Future Generation Networks (TAFGEN)-2015,978-1-4799-7315-6/15
25. W.Delishiya Moral, B.Muthu Kumar , "Improve the Data Retrieval Time and Security through Fragmentation and Replication in the Cloud ",International Conference on Advanced

Communication Control and Computing Technologies (ICACCCT)-2016,ISBN No.978-1-4673-9545-8
26. Mazhar Ali and Kashif Bilal, "DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security "IEEE Transactions on Cloud Computing, DOI 10.1109/TCC.2015. 2400460, 2015
27. Shripad Nadgowda, Sahil Suneja, Nilton Bila and Canturk Isci, "Voyager: Complete Container State Migration", IEEE 37th Inetrnational Conferenec on Distributed Computing Systems-2017,DOI 10.1109/ICDCS.2017.91
28. Shweta Alguje , Harish Patidar , Varsha Mishra, "Secure Replication in Cloud Computing Environment: A Survey",International Journal of Research In Science & Engineering Volume: 3 Issue: 2 March-April 2017,p-ISSN: 2394-8280
29. Steffen Kachele,Franz J. Hauck, "COSCAnet-FT: transparent network support for highly available cloud services", International Conference and Workshops on Networked Systems (NetSys)-2015,ISBN: 978-1-4799-5804-7
30. Marwa F. Mohamed, "Service replication taxonomy in distributed environments",Service Oriented Computing and Applications -2016,DOI 10.1007/s11761-015-0189-7