International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org || PP—112-118

A Critical Review on Quality of Service During Fault Tolerance In Cloud Computing

Bikash Chandra Pattanaik [1], Chandralekha [2]
[1](Computer Science and Engineering, BPUT, Odisha,India)
[2](Computer Science and Engineering, DRIEMS, Odisha, India)

ABSTRACT : *Cloud computing is an adoptable technology as it provides integration of software and hardware resources which are dynamically scalable. The dynamic environment of cloud results in various unexpected faults and failures. Fault tolerance enables a system to react gracefully to an unexpected equipment or programming malfunction. Here we are giving focus on reliability, fault tolerance and quality of service in cloud computing. The flexible and scalable property of dynamically fetching and giving up of computing resources in a cost-effective and device-independent manner with minimal management effort or service provider interaction the demand for Cloud computing paradigm has increased dramatically in last few years. Though lots of enhancement took place, cloud computing paradigm is still subject to a large number of system failures. As a result, there is an increasing concern among community regarding the reliability and availability of Cloud computing services. Dynamically provisioning of resources allows cloud computing environment to meet casually varying resource and service requirements of cloud customer applications. Quality of Service (QoS) plays an important role in the affective allocation of resources and has been widely investigated in the Cloud computing paradigm.*

KEYWORDS: *Cloud Computing, IoT, Mcc.*

I. INTRODUCTION

Internet has been a driving force towards the various technologies that have been developed. Arguably, one of the most discussed among all of these is Cloud Computing. Over the last few years, cloud computing paradigm has witnessed an enormous shift towards its adoption and it has become a trend in the information technology space as it promises significant cost reductions and new business potential to its users and providers. The advantages of using cloud computing include:

a) reduced hardware and maintenance cost,
b) accessibility around the globe, and
c) flexibility and highly automated processes wherein the customer need not worry about mundane concerns like software up-gradation.

Cloud Computing is an emerging trend to deploy and maintain software and is being adopted by the industry such as Google, IBM, Microsoft, and Amazon. Several prototype applications and platforms, such as the IBM ―Blue Cloud infrastructure, the Google App Engine, the Amazon Cloud, and the Elastic Computing Platform. Cloud Computing is perceived as the next progression that will impact organizational businesses and how they manage their IT infrastructures. The technology and architecture that cloud service and deployment models offer are a key area of research.

Even though there are numerous variations on the definition of Cloud Computing, some basic principles characterize this emerging computing paradigm. Cloud Computing provides technological capabilities— generally maintained off premises— that are delivered on demand as a service via the Internet. Given that a third party owns and manages public cloud services, consumers of these services do not possess resources in the cloud model but pay for them on a per-use basis. Thus virtualization of the resources is the key concept. In the real scenario, they are renting the physical infrastructure, platforms and applications within a shared architecture. Cloud offerings can vary from virtual infrastructure, computing platforms, centralized data centers to end-user Web-Services and Web applications to enormous other focused computing services.

II. DEPLOYMENT MODELS OF CLOUDS

In the cloud deployment model, networking, platform, storage, and software infrastructure are provided as services that scale up or down depending on the demand as depicted in figure 1. The Cloud Computing model has four main deployment models which are:

Private Cloud: Private cloud is a new term that some vendors have recently used to describe offerings that emulate cloud computing on private networks. It is set up within an organization's internal enterprise datacenter. In the private cloud, scalable resources and virtual applications provided by the cloud vendor are pooled together and available for cloud users to share and use. It differs from the public cloud in that all the cloud resources and applications are managed by the organization itself, similar to Intranet functionality. Utilization on the private cloud can be much more secure than that of the public cloud because of its specified internal exposure. Only the organization and designated stakeholders may have access to operate on a specific Private cloud. One of the best examples of a private cloud is Eucalyptus Systems.

Public Cloud: Public cloud describes cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web applications/web services, from an off-site third-party provider who shares resources and bills on a fine-grained utility computing basis. It is typically based on a pay-per-use model, similar to a prepaid electricity metering system which is flexible enough to cater for spikes in demand for cloud optimization. Public clouds are less secure than the other cloud models because it places an additional burden of ensuring all applications and data accessed on the public cloud are not subjected to malicious attacks. Examples of a public cloud include Microsoft Azure, Google App Engine.

Community Cloud: Infrastructure shared by several organizations for a shared cause and may be managed by them or a third party service provider and rarely offered cloud model. These clouds are normally based on an agreement between related business organizations such as banking or educational organizations. A cloud environment operating according to this model may exist locally or remotely. An example of a Community Cloud includes Facebook.

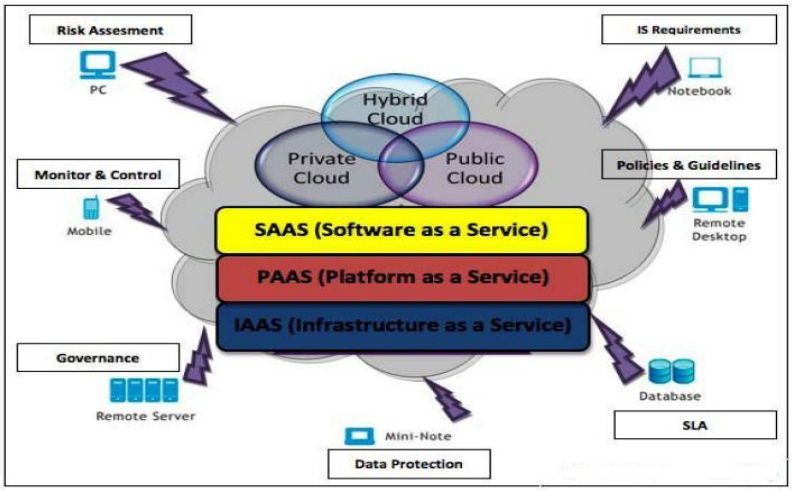


FIG. 1 CLOUD DEPLOYMENT MODEL

Moreover, with the technological advancements, we can see derivative cloud deployment models emerging out of the various demands and the requirements of users. A similar example being a virtual-private cloud wherein a public cloud is used in a private manner, connected to the internal resources of the customer's data-centre. With the emergence of high-end network access technologies like 2G, 3G, Wi-Fi, Wi-Max etc. and feature phones, a new derivative of cloud computing has emerged. This is popularly referred as ―Mobile Cloud Computing (MCC)‖. It can be defined as a composition of mobile technology and cloud computing infrastructure where data and the related processing will happen in the cloud only with an exception that they can be accessed through a mobile device and hence termed as mobile cloud computing. It's becoming a trend now-a-days and many organizations are keen to provide accessibility to their employees to access office network through a mobile device from anywhere.

III. SERVICE MODELS

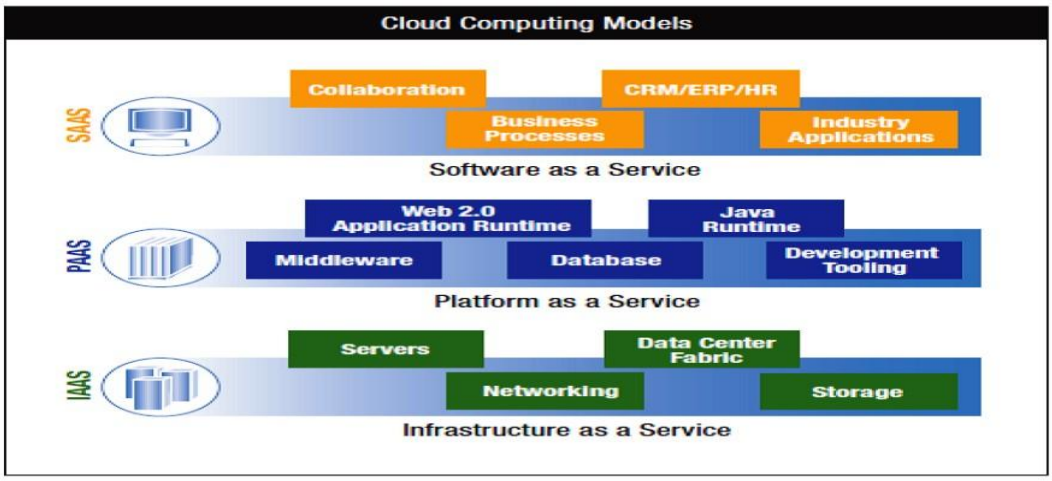
According to the different types of services offered, cloud computing can be considered to consist of three layers: software as a service (SaaS), platform as a Service (PaaS), and infrastructure as a Service (IAAS) (Iyer and Henderson, 2010; Han, 2010, Mell and Grance, 2010). Infrastructure as a Service (IaaS) is the lowest layer that provides basic infrastructure support service. Platform as a Service (PaaS) layer is the middle layer, which

offers platform oriented services, besides providing the environment for hosting user's applications. Software as a Service (SaaS) is the topmost layer which features a complete application offered as service on demand.

Software-as-a-Service (SaaS): SaaS can be described as a process by which Application Service Provider (ASP) provide different software applications over the Internet. This makes the customer to get rid of installing and operating the application on own computer and also eliminates the tremendous load of software maintenance; continuing operation, safeguarding and support. SaaS vendor advertently takes responsibility for deploying and managing the IT infrastructure (servers, operating system software, databases, data center space, network access, power and cooling, etc) and processes (infrastructure patches/upgrades, application patches/upgrades, backups, etc.) required to run and manage the full solution. SaaS features a complete application offered as a service on demand. In SaaS, there is the Divided Cloud and Convergence coherence mechanism whereby every data item has either the –Read Lock or –Write Lock. Two types of servers are used by SaaS: the Main Consistence Server (MCS) and Domain Consistence Server (DCS). Cache coherence is achieved by the cooperation between MCS and DCS. In SaaS, if the MCS is damaged, or compromised, the control over the cloud environment is lost. Hence securing the MCS is of great importance. Examples of SaaS includes: Salesforce.com, Google Apps.

Platform as a Service (PaaS): PaaS is the delivery of a computing platform and solution stack as a service without software downloads or installation for developers, IT managers or end-users. It provides an infrastructure with a high level of integration in order to implement and test cloud applications. The user does not manage the infrastructure (including network, servers, operating systems and storage), but he controls deployed applications and, possibly, their configurations. Examples of PaaS includes: Force.com, Google App Engine and Microsoft Azure.

Infrastructure as a Service (IaaS): Infrastructure as a service (IaaS) refers to the sharing of hardware resources for executing services using Virtualization technology. Its main objective is to make resources such as servers, network and storage more readily accessible by applications and operating systems. Thus, it offers basic infrastructure on-demand services and using Application Programming Interface (API) for interactions with hosts, switches, and routers, and the capability of adding new equipment in a simple and transparent manner. In general, the user does not manage the underlying hardware in the cloud infrastructure, but he controls the operating systems, storage and deployed applications. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis. Examples of IaaS include Amazon Elastic Cloud Computing (EC2), Amazon S3, GoGrid.



Combining the three types of clouds with the delivery models we get a holistic cloud illustration as seen in Figure 2, surrounded by connectivity devices coupled with information security themes. Virtualized physical resources, virtualized infrastructure, as well as virtualized middleware platforms and business applications are being provided and consumed as services in the Cloud. Cloud vendors and clients' need to maintain Cloud computing security at all interfaces.

IV. CLOUD COMPUTING ARCHITECTURE

Cloud computing can be divided into two sections, the user and the cloud. In most scenarios, the user is connected to the cloud via the internet. It is also possible for an organization to have a private cloud in which a user is connected via an intranet. However, both scenarios are identical other than the use of a private and public network or cloud. The user sends requests to the cloud and the cloud provides the service.

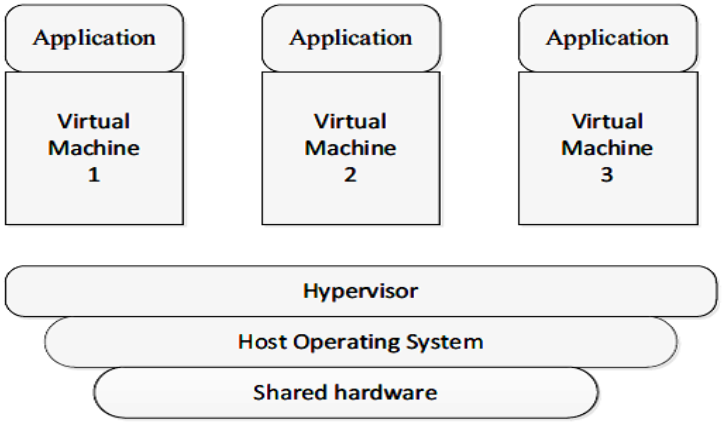


Fig. 3 Cloud Architecture

Within the cloud, a central server is responsible for administering the system and in many ways functions as the operating system of the specific cloud network. Another name for this is called ―middleware which is the central server for a particular cloud. Examples include Google App Engine and Amazon EC2.

V. RESEARCH CHALLENGES IN CLOUD COMPUTING

Cloud Computing research addresses the challenges of meeting the requirements of next generation private, public and hybrid cloud computing architectures, also the challenges of allowing applications and development platforms to take advantage of the benefits of cloud computing. The research on cloud computing is still at an early stage. Many existing issues have not been fully addressed, while new challenges keep emerging from industry applications. Some of the challenging research issues in cloud computing are given below.

- Service Level Agreements (SLA's)
- Cloud Data Management & Security
- Data Encryption
- Migration of virtual Machines
- Interoperability
- Access Controls
- Energy Management
- Multi-tenancy
- Server Consolidation
- Reliability & Availability of Service
- Common Cloud Standards
- Platform Management
- Fault Tolerance

VI. LITERATURE SURVEY

- WENBING ZHAO et al. (2010) proposed Low Latency Fault Tolerance (LLFT) Model that utilizes leader/follower replication approach and provides fault tolerance for distributed applications deployed within a cloud computing environment. The novel commitments of the LLFT middleware incorporate the low Latency Messaging Protocol, the leader-determined membership protocol and the virtual determinate Framework. [4]
- DAWEI SUN et al. (2013) put forward a dynamic adaptive fault tolerance strategy (DAFT) that is focused around the standards and semantics of cloud fault tolerance. An analysis on relationship between different failure rates and two different fault tolerance techniques, check-pointing and replication has been carried out. A dynamic adaptive model has been built by combining the two fault tolerance models which helps to

increase the serviceability. [28]

- ANJU BALA et al. (2014) put forward an idea of designing an intelligent task failure detection models for facilitating proactive fault tolerance by predicting task failures for scientific workflow applications. The working of model is distributed in two modules. In first module task failures are predicted with machine learning approaches and in second module the actual failures are located after executing workflow execution in cloud test-bed. Machine learning approaches such as naïve Bayes, ANN, logistic regression and random forest are implemented to predict the task failures intelligently from the dataset of scientific workflows.[34]
- HWAMIN LEE et al. (2009) proposed a fault tolerant and recovery system called FRAS system (Fault Tolerant and recovery Agent System). [1]This is an agent based system consisting of four types of agents. Recovery agent performs roll back recovery after occurrence of failure. Information agent hypothesis domain knowledge and information during a failure free operation. Facilitator controls the communication between agents and garbage collection agent performs garbage collection of data. Agent recovery algorithm is proposed to maintain a consistent state of a system and prevent domino effect.
- NAIXUE XIONG et al. (2007) Given that networks are dynamic and unexpected, Naixue-Xiong, investigates Failure detector properties with connection to real and programmed fault-tolerant cloud based network systems, in order to discover a general non-manual investigation strategy to self-tune corresponding parameters to fulfill user requirements.[24] Based on this general self-tuning method, they propose a dynamic and programmed Self- tuning Failure Detector scheme, called SFD, as an improvement over existing schemes.
- ANJALI MESHRAM et al. (2013) proposed fault tolerance model for cloud (FTMC). This model accesses the reliability of computing nodes and choses the node for the computation on the basis of reliability. The node can be removed if it does not perform well. [29]
- RAVI JAWAHAR et al. (2012): provided a new dimension for applications deployed in a cloud computing infrastructure which can obtain required fault tolerance properties from a third party. The model straightforwardly work fault tolerance solution to user's applications by combining selective fault tolerance mechanisms and discovers the properties of a fault tolerance solution by method of runtime monitoring.[20]
- SAGAR C JOSHI et al (2014) proposed a fault tolerance mechanism to handle server failures by migrating the virtual machines hosted on the failed server to a new location. Virtualization has been applied for data centers giving rise to the concept of virtual Data Centers (VDC) which have virtual Machine (VM) as the basic unit of allocation. Using appropriate resource allocation algorithms, multiple VDCs can be hosted on a physical data center. [31]
- SHIVAM NAGPAL et al (2013) proposed a fault tolerant model that takes decisions. Reliability of a node is estimated on the basis of 2 parameters; accuracy and time. If any of the nodes does not achieve the level then backward recovery is performed by the system. This model focuses on adaptive behavior of processing nodes and the nodes are removed or added on the basis of reliability.[36]
- SHUN-SHENG et al (2010) proposed Dual Agreement Protocol of Cloud Computing (DAPCC), keeping in consideration the scalable and virtual nature of cloud. DAPCC is proposed to tackle the agreement problem caused by faulty nodes which send wrong messages; it tells how the system achieves agreement in a cloud computing environment.
- HIEP NGUYEN (2013) proposes that one of the biggest challenges for diagnosing an abnormal distributed application is to pinpoint the faulty components. Black-Box online fault localization system called F-chain has been presented that can pinpoint faulty components immediately after a performance anomaly is detected. F-chain is presented as: a practical online fault localization system for large scale Iaas clouds. This system does not depend upon prior knowledge i.e. previously seen and unseen anomalies, and is practical for Iaas clouds. To achieve higher pinpointing accuracy, an integrated fault localization scheme has been introduced that consider both fault propagation patterns and inter component dependencies.[21]

VII. CONCLUSION

Cloud environment is dynamic which leads to unexpected system behavior resulting in faults and failures. In order to improve reliability and achieve robustness in cloud computing, failures should be assessed and handled effectively. Fault detection is one of the biggest challenges in making a system fault tolerant. This research work will provide a qualitative measure for identifying system fault-tolerance in terms of efficiency, robustness, availability, reliability and extent of fault management in dynamic environment for measuring the system performance. In order to achieve the objective "enhancement of QoS during fault tolerance in cloud computing" a comprehensive literature survey was carried out for cloud computing and various fault detection and fault tolerance techniques implemented in cloud computing. An extensive literature review was carried out for various models of artificial neural networks which can be used for fault detection. This research work will include an enhancement approach to increase the performance of cloud by optimizing difference QoS

parameters. Expected outcomes of the research work will be (1)Pro-active fault tolerance mechanism designed for dynamic clouds using Artificial Neural Network for fault detection can prove more beneficial than traditional models (2) measurement of detection time that is independent from the last heartbeat message, thus making the failure detector adaptive and increasing its accuracy (3)optimizing parameters like reliability, throughput etc. during fault management

REFERENCES

[1]. Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08* (pp. 1-10). Ieee.

[2]. Mell, P., & Grance, T. (2009). The NIST definition of cloud computing. *National Institute of Standards and Technology*, *53*(6), 50.

[3]. Plummer, D. C., Cearley, D. W., & Smith, D. M. (2008). Cloud computing confusion leads to opportunity. *Gartner Report*.

[4]. Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010, September). The characteristics of cloud computing. In *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on* (pp. 275-279). IEEE.

[5]. Carolan, S. J. (2009). Introduction to cloud computing. *Architecture. White Paper, 1st edn. Sun Microsystems (June 2009)*.

[6]. Furht, B. (2010). Cloud computing fundamentals. In *Handbook of cloud computing* (pp. 3-19). Springer US.

[7]. Kaushal, V., & Bala, A. (2011). Autonomic fault tolerance using haproxy in cloud environment. *Int. J. of Advanced Engineering Sciences and Technologies*, *7*(2), 54-59.

[8]. Patra, P. K., Singh, H., & Singh, G. (2013). Fault Tolerance Techniques and Comparative Implementation in Cloud Computing. *International Journal of Computer Applications*, *64*(14).

[9]. Bala, A., & Chana, I. (2012). Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing. International Journal of Computer Science Issues (IJCSI), 9(1).

[10]. Tchana, A., Broto, L., & Hagimont, D. (2012, March). Fault Tolerant Approaches in Cloud Computing Infrastructures. In *ICAS 2012, The Eighth International Conference on Autonomic and Autonomous Systems* (pp. 42-48).

[11]. Hayashibara, N., Defago, X., Yared, R., & Katayama, T. (2004, October). The φ accrual failure detector. In Reliable Distributed Systems, 2004. Proceedings of the 23rd IEEE International Symposium on (pp. 66- 78). IEEE.

[12]. Gupta, I., Chandra, T. D., & Goldszmidt, G. S. (2001, August). On scalable and efficient distributed failure detectors. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing* (pp. 170-179). ACM

[13]. Maier, G., Sommer, R., Dreger, H., Feldmann, A., Paxson, V., & Schneider, F. (2008, August). Enriching network security analysis with time travel. In *ACM SIGCOMM Computer Communication Review* (Vol. 38, No. 4, pp. 183-194). ACM.

[14]. Bahl, P., Chandra, R., Greenberg, A., Kandula, S., Maltz, D. A., & Zhang, M. (2007, August). Towards highly reliable enterprise network services via inference of multi-level dependencies. In *ACM SIGCOMM Computer Communication Review* (Vol. 37, No. 4, pp. 13-24). ACM.

[15]. Chen, W., Toueg, S., & Aguilera, M. K. (2002). On the quality of service of failure detectors. *Computers, IEEE Transactions on*, *51*(5), 561-580.

[16]. Bertier, M., Marin, O., & Sens, P. (2002). Implementation and performance evaluation of an adaptable failure detector. In *Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on* (pp. 354-363). IEEE.

[17]. Bertier, M., Marin, O., & Sens, P. (2003, June). Performance analysis of a hierarchical failure detector. In *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp. 635-635). IEEE Computer Society.

[18]. Défago, X., Urbán, P., Hayashibara, N., & Katayama, T. (2005, March). Definition and specification of accrual failure detectors. In *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on* (pp. 206-215). IEEE.

[19]. A Vouk, M. (2008). Cloud computing–issues, research and implementations.*CIT. Journal of Computing and Information Technology*, *16*(4), 235-246.

[20]. Jhawar, R., Piuri, V., & Santambrogio, M. (2013). Fault tolerance management in cloud computing: A system-level perspective. *Systems Journal, IEEE*, *7*(2), 288-297.

[21]. Huth, A., & Cebula, J. (2011). The Basics of Cloud Computing. *United States Computer*.

[22]. Brian, O., Brunschwiler, T., Dill, H., Christ, H., Falsafi, B., Fischer, M., ... & Zollinger, M. (2012). Cloud Computing. *White Paper SATW*.

[23]. Youssef, A. E. (2012). Exploring Cloud Computing Services and Applications.*Journal of Emerging Trends in Computing and Information Sciences*, *3*(6), 838-847.
[24]. Xiong, N., Vasilakos, A. V., Yang, Y. R., Qiao, C., & Andy, Y. P. (2012). A class of practical self-tuning failure detection schemes for cloud communication networks. *IEEE/ACM Transactions on Networking (ToN), submitted*.
[25]. Deng, J., Huang, S. H., Han, Y. S., & Deng, J. H. (2010, December). Fault-tolerant and reliable computation in cloud computing. In GLOBECOM Workshops (GC Wkshps), 2010 IEEE (pp. 1601-1605). IEEE.
[26]. Zhao, W., Melliar-Smith, P. M., & Moser, L. E. (2010, July). Fault tolerance middleware for cloud computing. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on (pp. 67-74). IEEE.
[27]. de Araújo Macêdo, R., & e Lima, F. R. L. (2004). Improving the quality of service of failure detectors with SNMP and artificial neural networks. In Anais do 22o. Simpósio Brasileiro de Redes de Computadores (pp. 583-586)
[28]. Sun, D. W., Chang, G. R., Gao, S., Jin, L. Z., & Wang, X. W. (2012). Modeling a dynamic data replication strategy to increase system availability in cloud computing environments. *Journal of computer science and technology*, *27*(2), 256-272.
[29]. Meshram, A. D., Sambare, A. S., & Zade, S. D. (2013). Fault Tolerance Model for Reliable Cloud Computing.
[30]. Nguyen, H., Shen, Z., Tan, Y., & Gu, X. (2013, July). FChain: Toward black-box online fault localization for cloud systems. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on* (pp. 21-30). IEEE.
[31]. Joshi, S. C., & Sivalingam, K. M. (2014). Fault tolerance mechanisms for virtual data center architectures. *Photonic Network Communications*, *28*(2), 154-164.
[32]. Wang, S. S., Yan, K. Q., & Wang, S. C. (2011). Achieving efficient agreement within a dual-failure cloud-computing environment. *Expert Systems with Applications*, *38*(1), 906-915.
[33]. Malik, S., & Huet, F. (2011, July). Adaptive Fault Tolerance in Real Time Cloud Computing. In *Services (SERVICES), 2011 IEEE World Congress on* (pp. 280- 287). IEEE.
[34]. Bala, A., & Chana, I. (2014). Intelligent failure prediction models for scientific workflows. *Expert Systems with Applications*.
[35]. Chandra, T. D., & Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM (JACM)*, *43*(2), 225-267
[36]. Mohsin Nazir *Department "Cloud Computing Overview : Current Research and Challenges" IOSR Journal of Computer Engineering (IOSR-JCE) ISSN: 2278-0661, ISBN: 2278-8727Volume 8, Issue 1 (Nov. - Dec. 2012), PP 14-22*
[37]. Amin Zeeshan, Sethi Nisha, Harshpreet Singh "Review on Fault Tolerance Techniques in Cloud Computing" *International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 18, April 2015*