International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 08 | Nov-2015 www.irjet.net p-ISSN: 2395-0072

A Comparative Review on Fault Tolerance methods and models in Cloud Computing

Vinay Kumar[1], Dr. Sandeep Sharma[2]

[1] M.Tech Scholar, Computer Engineering and Technology Department, Guru Nanak Dev University, Punjab, India
[2] Prof, Computer Engineering and Technology Department, Guru Nanak Dev University, Punjab, India

---***---

Abstract - ***Cloud computing research area has just come up as a latest prototype for delivering, hosting services, in which common utilities (CPU, Storage) are provided to users which can be rented and freed in an on-demand manner over the internet. Even though, it provides various characteristics like on-demand supply of utilities or resources, multi-tenancy, decreased cost, agility etc., and also associated flaws and risks with it. A variety of research issues areas are associated with it and fault tolerance (FT) is one of them. It is the procedure of detecting failures and faults and if a fault takes place due to the hardware/software failure afterward the cloud computing system must also perform correctly. In real time applications, delay in processing due to fault is not accepted in cloud. By using virtualization technique, high availability of resources with minimum down time is achieved. To facilitate this feature a method is used to discover failover in physical servers which further detects failure in the host. FT is essential for the system to assurance both guaranteed availability & continuous reliability of critical application and services execution. So in this, robust Fault Tolerant (RoFT) system is required. To understand fault tolerance in it, it is necessary to know more about various types of failure/faults. Our emphasize in this paper is on essential FT concepts by knowing its policies namely; RFTP (Reactive), PFTP (Proactive) and some related procedures or methods apply on different failures or faults. A lot of research on different FT frameworks, algorithms, methods that are implemented, designed by professional has been accomplishing.***

Key Words: *Cloud Computing, PFTP, Fault Tolerance, RFTP.*

1. INTRODUCTION

"Cloud Computing (CC) is basically an on-demand delivery of Information Technology (IT) resources or subscription-based services through the internet with (PAUG) pay-as-you-go model or pricing". In other words, CC is a running programs over internet instead of local computer, storing/retrieving data over internet instead of local computer hard disk, renting resources over the internet rather than buying them on your own. Several cloud computing service providers (SPs) delivers cloud computing services around the world with the help of various SPs including Microsoft, Google, IBM 2, and Yahoo are quickly deploying data centers (DCs) in various locations. Modern DCs have thousand of servers connected internally to each other and numerous applications hosted on these servers. Most frequently, these large DCs are not real while virtual, and numerous resources of computing are offered as configurable VMs (Virtual Machines) to individual user over the Internet.

The Fig-1 gives a cloud computing overview and various services offered by it. Some particular services are recognized as Infrastructure (IaaS), Platform (PaaS), Software (SaaS), Monitoring (MaaS), Communication (CaaS), Hardware (HaaS), Anything or Everything as a

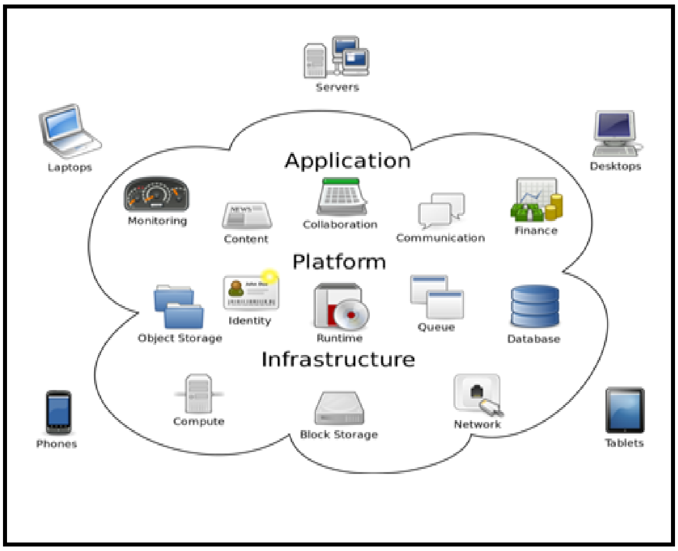

Fig -1: CLOUD COMPUTING OVERVIEW

Service (XaaS) respectively. But cloud offers main three services models : IaaS, PaaS and SaaS**.** In *IaaS*, the ability offered by cloud to the customer is to supply storage, processing, networks and additional resources of computing by using these customer is able to run and deploy any kind of software which contain different types of applications and also operating systems (OSs). Examples of Infrastructure as a Service providers include GoGrid, Flexiscale and Amazon EC2. In *PaaS*, it offers PL (platform layer) resources, including software development frameworks and operating system support. It is offered service to App developers so they can make application in an easy way without buying or managing underlying development tools and large servers. Examples of Platform providers comprise Force.com, Microsoft Windows Azure and Google App Engine. In *SaaS*, offers on-demand applications and in which domain related applications are develop, operate and host by cloud providers that can be used by customers through Internet on a PAUG basis. Examples of Software as a service providers include Youtube, Facebook, Rackspace, SAP Business ByDesign and Salesforce.com.to provide software as a service. Other examples include online word processing and spreadsheet tools, Gmail, WhatsApp, and SAP.

1.2 Cloud computing characteristics

- Multi-tenancy services or facilities owned by various SPs are located in a one DC. The issues related with management and performances are shared between infrastructure provider and SPs of these services.
- Shared Resource Pooling allows dynamically assigned and reassigned different virtual and physical IT resources, according to cloud service user demands.
- Resources are accessible over the internet or network, supporting heterogeneous user platforms such as a laptop, mobile phone, or a PDA, is capable to access services of cloud.
- Providers provide its services like IaaS, PaaS and Saas according to the SLA stipulated with its customers.
- High agility that allows SPs to react quickly to quick changes according to customers own needs or in service demand.
- Dynamic resource provisioning (DRP) enables SPs to attain resources or utilities on the basis of present demand and significantly lower the OC (operating cost).
- Utility-based pricing reduces service OC because they charges consumers on a PAUG basis [1].

Cloud Computing have some Commercial Products like, *First*, Amazon (EC2) Elastic Compute Cloud allows CC consumers to manage as well as launch instances of server in DCs using available utilities, APIs or tools. *Second*, Microsoft Windows Azure (MWA) applications can be used by executing on local systems and executing in the cloud. *Third,* Google App Engine (GAE) platform is used for conventional web applications managed by Google DCs. At present, Java and Python programming languages are supported, and Web frameworks include CherryPy, Django, Pylons that run on the GAE, and web2py, a customize G-written web apps framework related to Java Server pages (JSP), ASP.NET [2].

The structural design have four layer in CC environment namely; application, platform, infrastructure and Data center/Hardware. These four layers need various levels of FT methods to offer flawless service. In CC failures that can arise classified into 2 classes namely; Data Failures (DF) and Computation Failures (CF) [4]:

1) DF: It includes failures due to missing source information, other flaws in the data and corruption of information or data.

2) CF: It includes failures of all types like storage access exception, faulty or slow Virtual machines, etc of infrastructure or hardware failures.

A variety of research issues areas are associated with it and fault tolerance (FT) is one of them. FT is essential for the system to assurance both availability & reliability [3] of critical application and services execution. To minimize the failures impact on application execution, its occurrence in the system and it should be handled by appropriate technique. According to studies conducted, in cloud mostly faults arise due to failures of *hardware* primarily in hard disk (HD), processors, memory, and IC (integrated circuit) sockets. It has been calculated from studies, after every few minutes a cloud computing system with 100000 processors will face a processor failure. Some *software* faults responsible for failure of application and congestion in network, faults in network because of server overload etc. and that forbids communication among the end users and cloud. For that reasons, it is essential in cloud computing for experienced FT scheme that manages faults in a various ways.

This review paper is arranged as succeeds. In section II explains FT taxonomy, Section III we review the work associated to FT & draws different FT models analytical comparison, and to end with in Section IV we here some concluding explanations'.

2. FAULT TOLERANCE OVERVIEW

It is the procedure of detecting failures and faults and if a fault takes place due to the hardware/software failure afterward the cloud computing system must also perform correctly. For reliable cloud computing failures should be managed in an efficient way. It will also guarantee availability and robustness. In cloud computing failure of single node may affect the result of the entire system. In real time applications, delay in processing due to fault is not accepted in cloud. By using virtualization technique, high availability of resources with minimum down time is achieved. To facilitate this feature a method is used to discover failover in physical servers which further detects failure in the host. FT system might be capable to bear several types of faults containing- transient, permanent hardware faults or intermittent, software and design errors, damage physically, or operator errors.

FT concern with every unavoidably procedures to facilitate the some features like robustness and dependability. Robustness is related to giving an accurate supply in an unfavorable condition originating because of an unpredictable system state [3]. Dependability is interrelated with several Quality of service (QOS) features offered by cloud computing system, or it's something that necessitate to be achieved, it include the attributes like availability and reliability. To implement FT in CC include some benefits improved performance metrics, lower cost, recovery from failure [5].

2.1 Fault tolerance Taxonomy

In Cloud, two general FT policies i.e. PFTP (Proactive FT Policy) and RFTP (Reactive FT Policy) are shown in Fig-2. Several methods are used to offer FT by using these policies.

2.1.1 PFTP

PFTP means early prediction of the failures before it actually arises. The principle of PFTPs is to avoid spare effort for recovering the failed nodes, jobs, by predicting failures early and swap proactively the uncertain elements with additional functioning elements. By the real time systems, PFT systems are able to accomplish the time constraints set [5]. Some PFT techniques are as follows:

1) PFTP Using Software Rejuvenation: Software Rejuvenation is the PFT technique in which an application is instantly terminated and then restarted as a fresh or clean state. In this technique, scheduled or programmed repeated reboots and then system restart or resumes with a fresh position after every reboot.

2) PFTP Using Self Healing: Self Healing is the PFT technique used where multiple different VMs are executing various instances of one application and when fault occurs, the failure of instances of different application

can be handled automatically by using self healing [4].

3) PFTP Using Preemptive Migration: Preemptive Migration is the PFT technique in which the parts of an application are migrated to different computing node before real fault occurs. Preemptive migration relies on control system known as feedback-loop(F-L) and by using this technique every application are continuously analyzed and controlled [5].

2.1.2 RFTP

RFTP handles the failure or fault. It helps to minimize the impact of failures or faults when it really happens on the execution of application in cloud. FT techniques however provide good solution for common computing environment; it cannot accomplish the time constraint set by the real time computing systems. Some RFT techniques are as follows [5]:

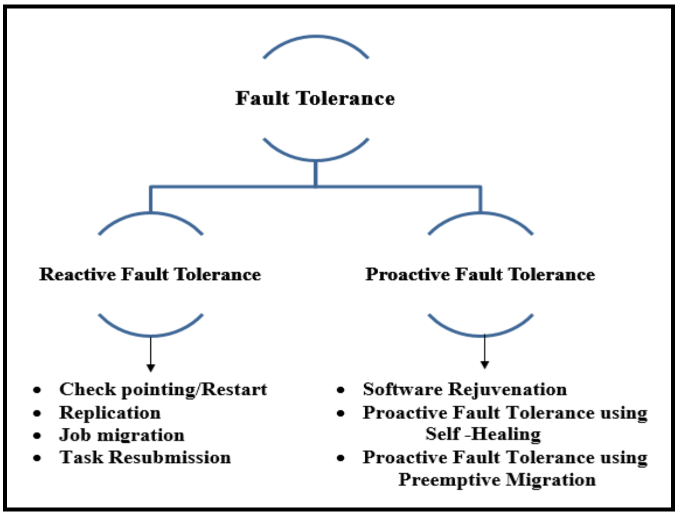

Fig -2: FAULT TOLERANCE TECHNIQUES

1) Replication: It is the way of holding many replicas of object or data. Using this procedure, consumer demands for a replica from a collection of copies executed by various resources till the job/task is crashed or finished. It is procedure to add duplication in a cloud. To preserve stability by all of replica, degree of replication, management of replica etc. Hadoop, AmazonEC2 and HAProxy tools are used to offer this method in a cloud. A replication protocol (RP) offers consistency among copies of identical data. If only one copy of replicas is changed by a consumer then consistency issue can be raised and also, as well as number of copies increases, expense of preserving or controlling the stability will also rises.

2) Checkpointing/Restart: When task scheduling is doing then checkpoints are inserted to recognize fault incidence. When actually a failure occurs, Checkpointing/Restart techniques take less time and less computation as a result of task or application is begin where previous checkpoint left off to position of failure or fault instead of rebooting of task or application from initial position.

3) Job Migration: When due to machine or resource failure any task or job fails then the task is moved on another virtual machine (VM) where it continues its execution [4].

4) *Task Resubmission:* Task resubmission (TR) means that when any task or job fails then it is submitted again to the same VM or to some other VM or resource without disturbing cloud computing system workflow at a runtime [5].

2.1.3 Versatile or Adaptive Fault tolerance

All process has done automatically according to the circumstances [6].

Both PFTPs and RFTPs have benefits and drawbacks. Some experimentation outcomes which show that (RFTP) Checkpointing/Restart scheme are less capable or efficient than (PFTP) preemptive migration procedure. Even if proactive techniques are much efficient, that's why it is not usually compared with RFTP methods. By incorrect or false predictions due to PFT, system is very less affected. At the time of development, fault tolerance methods are not useful as reactive techniques are quite easy to employ and also it might not be appropriate where systems require greater amount of clusters or virtual machines (VMs) availability because availability decreases severely, once a failure appears.

3. RELATED WORK

Pranesh Das [7] et al proposed a model VFT (Virtualization FT) to increase the availability of system and to decrease service time of the system. RFT method of VFT model resides of two modules, named as a Decision Maker and a Cloud Manager. Both modules are used for managing the faults, to handle the load balancing (LB) and virtualization. In first step, it comprises LB as well as virtualization. In second step, FT is accomplished by fault handler (FH), checkpointing and redundancy. In this approach a FH is integrated with virtualization segment and it chokes faulty nodes (FNs) for future requests (FRs) that are not recoverable and hold down its virtual nodes (VNs) and eliminates unstable faults in software and makes VNs accessible for FRs from the improvable FNs.

Ravi Jhawar et al [8] initiate innovative, system-level, standard point of view to mange and create FT in CC system. They proposed a high level comprehensive technique to protecting accomplishment information's of FT methods to users and app developers with the help of appropriate Service Layer (SL). This SL acknowledges the customer to specify preferred stage of FT, and information regarding FT methods which are resides in CC are not available.

Ifeanyi P. Egwutuoha et al [9] offers new flexibility and capacity solution with facilities of many VMs to HPC (High performance computing) for computational demanding applications. Fault tolerance (FT) allows high performance computing systems on cloud with multiple VMs or nodes to complete execution in the present of fault. FT techniques used for HPC is restart/checkpoint.

Deepak Poola [10] et al designed a new algorithm based on scheduling, to reduce cost of execution to submitted jobs on CC resources by using on-demand and spot instances pricing models while reaching the workflow (WF) time limit. It uses checkpointing technique and save cost to 14%.

Baoyan Song et al [11] talk about in cloud computing frameworks generally produce lot of intermediate data (ID) that are small and essential for job termination. They proposed two ID based fault-tolerant algorithms, correspondingly the outer and inner task IDF. Proposed algorithms results represent, when there are server failures, it maintains the reliability of the system.

Bipin B. Nandi et al [12] proposed FTaaS that can offer temporal and spatial repetitions. In paper, they examine a universal perspective, where an occupant (who explore FT services) be able to perform different FT condition, with various modes and the task of the provider (who advertises FT services) can constitute requirements according to the tenants', it satisfy his clients and to maximize profits earnings.

Ravi Jhawar et al [13][24] proposed a way to assess FT components so as to apply virtualization which can undoubtedly increase availability as well as the reliability of operations utilized in various VMs. They studied FT components to build up a method to select and identify components that equivalent customer's FT specifications in various deployment manners.

Sheheryar Malik et al [14] proposed a real time FT model in CC. To make judgment in accordance to the reliability of the virtual machines (VMs) that is processing nodes and then tolerates the system faults. The VMs reliability alters in each CC (computing cycle). When VM generate an accurate results in a specify time period, then that VM reliability increases, otherwise its reliability decreases. CC systems will do safety measures (SM) or backward recovery (BR) if VM does not attain minimum reliability level.

Wenbing Zhao et al [15] proposed a middleware in LLFT (Low Latency FT) which uses L/F replication technique and offers FT to distributed applications executed in data center. LLFT middleware resides Leader-Determined Membership Protocol (LDMP), a Virtual Determinizer (VD) and a Low Latency Messaging Protocol (LLMP) Framework. The Membership Protocol (MemP) offers rapid recovery service and reconfiguration while replica turns out to be inadequate as well as it leaves or joins a set. A LLMP offers reliable delivery among replicated systems.

Anjali D.Meshram et al [16] proposed a FTMC model. By considering reliability parameter, it tolerates the faults of each computing node. Mostly remote computers are used for processing so there is probability of occurrence of more errors. For this reason, CC environment must be FT and also there is a need of precise scheduling to executing tasks.

Oren Laadan et al [17] talks about ASSURE, that come out with RP (rescue point) method. When a fault or failure occurs at a random location in the program, then Assure restored execution to a suitable RP and causes the program to get back execution by virtualizing the programs with existing fault-handling or error-handling facilities.

Gang Chen et al [18] discuss about SHelp, implements rescue points (RPs) approach with weight. It is another independent fault tolerance (FT) system proposed which uses checkpointing/restart as fault tolerance method in virtual environment.

Jayadivya S K et al [19] proposed model FTWS (Fault Tolerant Workflow Scheduling) offering FT with the help of tasks priority by using tasks resubmission and replication in an analytical metric. To meet the deadline, this model replicates tasks and schedule tasks.

K. Ganga [20] et al talks about the various FT methods and give more attention how replication of tasks is performed in CC on the basis of scientific workflow.

Naixue Xiong [21] et al introduces a system SFD that gives information about how to detect faults in CC. It can regulate control parameters of fault detection and also confirms improved fault detection.

Min Lu [22] et al in this paper, they analyzed hybrid cloud performance and behavior using model named like Queuing Petri nets. During resource provision stage, it recaptures failure occurring in virtual node by constructing a new FT policy.

Alain Tchana et al [23] proposed a FT scheme in which faults related to application are found and fixed at consumer stage and the faults related to hardware and VM are found and fixed at provider stage. For recovered virtual machines, they generate restore points by using Checkpointing procedure.
Liqiong Chen [25] et al proposed a CFN model, so CC Fault Net model is generated various components of CC like failure process, detection process, and service resources and so on. Various components of CC are created by using Petri net that can incorporate dynamically into proposed model.

JiSu Park et al [26] offers a monitoring method in relation to FT and pays more attention on mobile CC (MCC). A monitoring scheme is made on the basis of model named; Markov chain to gather information of state that is required to analyze reliability of FT in MCC.

Jorge G. Barbosa [27] et al talks related to RT and MT task execution algorithms and their purpose is to decrease the faults as well as improving efficiency of power in CC. To request for carrying out a task, a first algorithm which set aside the essential resources necessary for task successfully done in its time limit. Whereas, a second algorithm offers complete resources access and specific task have right to use utmost essential resources at least amount of time that are essential for task successfully done.

To analyze various *Literature Gaps* there is necessitate applying Autonomic FT (AFT) with the help of various parameters in cc domain. In CC, various problems faced in FT at the time of papers review are as follows:

- Its complex to understand the altering state of the system because CC domain is dynamically scalable, unpredicted and is frequently offered services as virtualized resources.
- Partial detail is offered to consumers as of large difficulties in system, therefore it's hard to propose a best possible FT solution.
- Prediction framework and Fault Monitoring requirements for applications which are executing in real-time in CC domain must develop.

4. COMPARATIVE ANALYSIS OF FAULT TOLERANCE MODELS

On the basis of some metrics acquired from these fault tolerance models, a logical comparison is done. An existing FT techniques in cloud computing consider some certain fault tolerance properties or attributes or features are apply logically to figure out all fault tolerance models. In above, Table-1 depicts the logical comparison along with various fault tolerance models on the basis of above mentioned properties/parameters in a table. Some parameters are explained in brief below:

Table -1: Comparative Analysis of Fault Tolerance Models

SR. NO.	FAULT TOLERANCE MODEL	FAULT TOLERANCE (FT) PROPERTIES				
		REACTIVE FT	PROACTIVE FT	RELIABILTY	PERFORMANCE	RESPONSE TIME
1	VFT	YES	YES	HIGH	HIGH	HIGH
2	LLFT	YES	NO	HIGH	HIGH	AVERAGE
3	FTM	YES	NO	HIGH	LOW	AVERAGE
4	ASSURE	NO	YES	AVERAGE	HIGH	AVERAGE
5	SHelp	YES	NO	LOW	AVERAGE	HIGH
6	FTMC	YES	NO	HIGH	LOW	AVERAGE
7	AFTRC	NO	YES	HIGH	HIGH	AVERAGE
8	FTWS	YES	NO	AVERAGE	AVERAGE	AVERAGE

1) Types of fault tolerance method – that may be reactive FT or proactive FT.

2) Reliability - that aims to deliver precise or suitable results in a period of time bounded situation.

3) Performance – checking the system effectiveness or efficiency. System performance must improve with some reasonable cost; for example response time may be decrease by approving adequate stalls.

4) Response Time (RT) – time acquired by a specific technique or model or algorithm or procedure to respond and its value should be smallest or minimum.

5. CONCLUSION

Cloud Computing has turned into a trendy computational technology across the world. Its enormous supports have offered services to make sure guaranteed availability and continuous reliability. For this reasons, its necessitate for having experienced FT method which manages faults as well as failures in various aspects. We focus on general FT approaches in CC in this paper. As we know, it is a current key research area and a lot of experimentation efforts are accomplished in this area, particularly in developing a separate FT scheme. In this area, there are several FT procedures designed by professionals. Our final goal is to examine all fault tolerance techniques and to identify and have knowledge about the drawbacks and to build up a fault tolerance scheme that can handle every faults type in various manners.

REFERENCES

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," The Brazilian Computer Society Conference on cloud computing, Springer, pp. 7-18, 2010.
[2] G. Shroff, "Enterprise Cloud computing technology, architecture, applications," Cambridge south asian ed., pp. 51-60, 2011.
[3] A. Bahga, and V. Madisetti, "Cloud computing A hands-on approach," UNIVERSITIES PRESS, 1st ed., pp. 117-120, 2014.
[4] A. Ganesh, Dr. M. Sandhya, and Dr. S. Shankar, "A study on Fault Tolerance methods in cloud computing," IEEE International Conference on Advance Computing Conference (IACC), pp. 844-849, 2014.
[5] D. Mittal, and N. Agarwal, "A Review Paper on Fault Tolerance in Cloud Computing," IEEE International Conference on Computing for Sustainable Global Development (INDIACom), pp. 31-34, 2015.
[6] A. Mahalkari, and Prof. R. Tondon, "A Replica Distribution based Fault Tolerance Management for Cloud Computing," International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5, no. 5, pp. 6880-6887, 2014.
[7] P. Das, and P. M. Khilar, "VFT: A virtualization and fault tolerance approach for cloud computing," IEEE conference on Information & Communication Technologies (ICT), pp. 473-478, 2013.

[8] R. Jhawar, V. Piuri, and M. D. Santambrogio, "Fault Tolerance Management in Cloud Computing: A System-Level Perspective," IEEE International Systems Journal, Vol. 7, no. 2, pp. 288-297, June 2013.
[9] I. P. Egwutuoha., S. Chen, D. Levy, and B. Selic, "A Fault Tolerance Framework for High Performance Computing in Cloud," IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp. 709-710, May 2012.
[10] D. Poola, K. Ramamohanarao, and R. Buyya, "Fault-Tolerant Workflow Scheduling Using Spot Instances on Clouds," 14th ELSEVIER International Conference on Computational Science (ICCS), pp. 523-533, 2014.
[11] B. Song, C. Ren, X. Li, and L. Ding, "An Efficient Intermediate Data Fault-Tolerance Approach in the Cloud," 11th IEEE Conference on Web Information System and Application, pp. 203-206, 2014.
[12] B. B. Nandi, H. S. Paul, A. Banerjee, and S. C. Ghosh, "Fault Tolerance as a Service," 6th International Conference on Cloud Computing, pp. 446-453, IEEE, 2013.
[13] R. Jhawar, V. Piuri, and M. D. Santambrogio, "Fault tolerance management in IaaS clouds," IEEE 1st AESS European Conference on Satellite Telecommunications (ESTEL), pp. 1-6, 2012.
[14] S. Malik, and F. Huet, "Adaptive Fault Tolerance in Real Time Cloud Computing," IEEE World Congress on Services (SERVICES), pp. 280-287, 2011.
[15] W. Zhao, P. M. Melliar-Smith, and L. E. Moser, "Fault tolerance middleware for cloud computing," 3rd International Conference on Cloud Computing, pp. 67-74, IEEE, 2010.
[16] A. D. Meshram, A. S. Sambare, and S. D. Zade, "Fault Tolerance Model for Reliable Cloud computing." International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), vol.1, no.7, pp. 600-603, July 2013.
[17] S. Sidiroglou, O. Laadan, C. Perez, N. Viennot, J. Nieh, and A. D.Keromytis, "Assure: automatic software self-healing using rescue points," ACM Sigplan Notices vol. 44, no. 3, pp. 37-48, 2009.
[18] G. Chen,, H. Jin, D. Zou, B. B. Zhou, W. Qiang, and G. Hu, "SHelp: Automatic Self-healing for Multiple Application Instances in a Virtual Machine Environment," International Conference on Cluster Computing (CLUSTER), pp. 97-106, IEEE, 2010.
[19] S. K. Jayadivya, S. J. Nirmala, and M. S. Bhanu, "Fault tolerant workflow scheduling based on replication and resubmission of tasks in Cloud Computing," International Journal on Computer Science and Engineering (IJCSE), Vol. 4, no. 6, pp. 996-1006, 2012.
[20] K. Ganga, and S. Karthik, "A fault tolerent approach in scientific workflow systems based on cloud computing," IEEE International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME), pp. 387-390, 2013.
[21] N. Xiong, A. V. Vasilakos, J. Wu, Y. R. Yang, A. Rindos, Y. Zhou, W. Z. Song, and Y. Pan, "A self-tuning failure detection scheme for cloud computing service," IEEE 26th International Conference on Parallel & Distributed Processing Symposium (IPDPS), pp. 668-679, 2012.
[22] M. M. Lu, and H. Yu, "A Fault Tolerant Strategy in Hybrid Cloud Based on QPN Performance Model," International Conference on Information Science and Applications (ICISA), pp. 1-7, IEEE, 2013.
[23] A. Tchana, L. Broto, and D. Hagimont, "Approaches to cloud computing fault tolerance," International Conference on Computer, Information and Telecommunication Systems (CITS), pp. 1-6, IEEE, 2012.
[24] R. Jhawar, V. Piuri, and M. D. Santambrogio, "A comprehensive conceptual system-level approach to fault tolerance in cloud computing," IEEE International Systems Conference (SysCon), pp. 1-5, Mar 2012.
[25] G. Fan, H. Yu, L. Chen, and D. Liu, "Model Based Byzantine Fault Detection Technique for Cloud Computing," IEEE Asia-Pacific Services Computing Conference (APSCC), pp. 249-256, 2012.
[26] J. Park, H. C. Yu, K. S Chung, and E. Y. Lee, "Markov chain based monitoring service for fault tolerance in mobile cloud computing," IEEE Workshops of International Conference on, Advanced Information Networking and Applications (WAINA), pp. 520-525, 2011.
[27] A. M. Sampaio, and J.G. Barbosa, "Dynamic Power- and Failure-Aware Cloud Resources Allocation for Sets of Independent Tasks," IEEE International Conference on Cloud Engineering, pp.1-10, 2013.

BIOGRAPHIES

Vinay Kumar pursuing M. Tech in Computer Science & Engineering from Department of Computer Engineering & Technology, Guru Nanak Dev University, Amritsar, Punjab, India. His research interests include Cloud Computing.

Dr. Sandeep Sharma is working as professor in the Department of Computer Engineering & Technology, GNDU, Amritsar, Punjab, India. He has published many research articles in the National/International conferences. His research interests include Cloud Computing and Parallel Processing.