Survey on Hybrid Cloud With Fault Tolerance

Mrs.Sowmya C.N.[1], Prof. Janardhan Singh[2]
[1]Dept of CSE
[2]Asociate Professor, Dept of CSE
[1, 2] Cambridge Institute Of Technology , Bangalore, India

Abstract- *Cloud computing has brought about a transformation of information technology from a product to a service. It has enabled the availability of various software, platforms as well infrastructural resources as scalable services on demand over the internet. However, the performance of cloud computing services is hampered due to their inherent vulnerability to failures owing to the scale at which they operate. It is possible to utilize cloud computing services to their maximum potential only if the performance related issues of reliability and throughput are handled effectively by cloud service providers. Therefore, fault tolerance becomes a critical requirement for achieving high performance and reliable in cloud computing. This paper presents a comprehensive overview of fault tolerance-related issues in cloud computing. The objective is to provide insights into the existing fault tolerance approaches as well as challenges yet required to be overcome. The survey enumerates a few techniques that may be used for efficient solutions and also, identifies important research directions in this area.*

Keywords- Cloud Computing, Fault Tolerance, Scientific Workflows, Hybrid cloud, Checkpoint, Cloud, Dependability, Replication.

I. INTRODUCTION

Cloud Computing provides virtual infrastructure to the companies and reduce their load of maintaining backend architecture. There are many other benefits cloud to its users. Most of the people use public cloud but that comes with a compromise with performance or security. It makes companies switch to the private cloud and enjoy the benefits of cloud computing. But nowadays, organizations are more likely moving towards – the hybrid cloud. In this survey paper, we will explore what is hybrid cloud computing, hybrid cloud architecture, hybrid cloud service providers, hybrid cloud challenges, and finally find a conclusion if hybrid cloud is a good choice.

Fault tolerance in **cloud computing** is about designing a blueprint for continuing the ongoing work whenever a few parts are down. This helps the enterprises to evaluate their infrastructure needs and requirements, and **provide** services when the associated devices are unavailable.

Main Concepts behind Fault Tolerance in Cloud Computing System

Replication: The fault-tolerant system works on the concept of running several other replicates for each and every service. Thus, if one part of the system goes wrong, it has other instances that can be placed instead of it to keep it running. Take, for example, a database cluster that has 3 servers with the same information on each of them. All the actions like data insertion, updates, and deletion get written on each of them. The servers, which are redundant, would be in inactive mode unless and until any fault tolerance system doesn't demand the availability of them.

Redundancy: When any system part fails or moves towards a downstate, then it is important to have backup type systems. For example, a website program that has MS SQL as its database may fail in between due to some hardware fault. Then a new database has to be availed in the redundancy concept when the original is in offline mode. The server operates with the emergency database which comprises several redundant services within.

II. ENHANCING FAULT TOLERANCE OF CLOUD NODES USING REPLICATION TECHNIQUES.

The cloud technologies are gaining boom in the field of information technology, on the same side cloud computing sometimes results in failures. These failures demand more reliable frameworks with high availability. The request made by the user is replicated and stored in various VMs. If one of the VMs fail, the other can respond to increase the reliability. A lot of research has been done and being carried out to suggest various schemes for fault tolerance thus increasing the reliability. Earlier schemes focus on only one way of dealing with faults but the scheme proposed by the author in this paper presents an adaptive scheme that deals with the issues related to fault tolerance in various cloud infrastructure. The projected scheme uses adaptive behavior during the selection of replication and fine-grained check pointing methods for

attaining a reliable cloud infrastructure that can handle different client requirements.

III. HYBRID FAULT TOLERANCE MODEL FOR CLOUD DEPENDABILITY

Our solution is a hybrid fault tolerance strategy that combines checkpoint and replication strategies to ensure Cloud resources availability. We prove that this combination outperforms, in terms of resources additional workload, the other proposed strategies. We prove that this combination outperforms, in terms of resources additional workload, the other proposed strategies. Cloud resources architecture that has been used to test the different fault tolerance strategies is detailed in Fig. 1

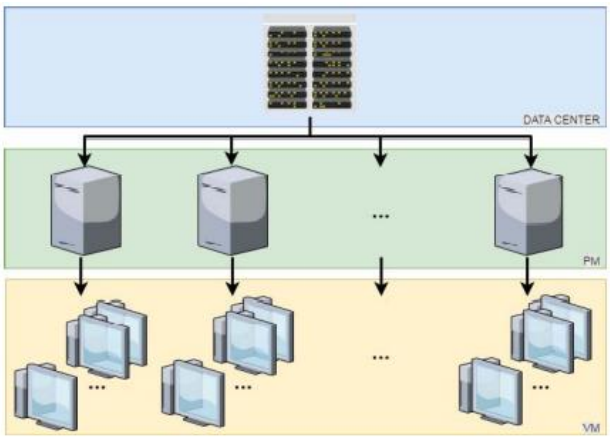

Fig1:cloud architecture.

Cloud architecture our strategy focuses only on the configuration of virtual and physical machines regardless of the characteristics of DCs that host these machines, in our Cloud, only one DC is used. In this DC there are 50 PM, each has a RAM of 16 GB and a various number of processing elements with 500 MIPS each of them. To execute user applications, we create in this Cloud VMs.

By tolerating faults in Cloud resources, the reliability of Cloud applications and services can be highly improved. Using traditional strategies maintains the continuity of Cloud services, but adds extra workload and execution time, which affects the quality of Cloud services as well as service fees. The evaluation results have shown that the presented strategy allows, on the one hand, to ensure availability and reliability of Cloud applications, and on the other hand, to decrease Cloud infrastructure workload and execution time.

IV. APPROACH FOR SELECTING AND INTEGRATING CLOUD SERVICES TO CONSTRUCT HYBRID CLOUD

In this approach combine all your researched information in form of a journal or research paper. In this researcher can take the reference of already accomplished work as a starting building block of its paper. proposed a C-SIP and a hybrid CSB, which are automation solutions that support hybrid cloud deployment. The C-SIP consists of a cloud service selection method that chooses a combination of various clouds according to user's requirements and a script generation method to solve the complexity problem between clouds. In the selection technique, we refer to the reference patterns and CDPs provided by major public cloud vendors such as AWS and Azure. We defined a generalized pattern and proposed a method for selecting patterns, suggesting that the service derived from the selection result is bound to a pattern. Finally, the accuracy of the proposed pattern selection method for evaluation was measured through comparison with the architecture in the AWS casebook. We proposed a combination of cloud services with customized patterns, so that users can simply input the purpose and requirements of building the cloud services. Script generation techniques that support cloud-to-cloud integration include a process in which a user determines the elements of a script, as well as a model that supports executable code-level script generation. The integration script generation model has a structure that can easily expand the services and methods belonging to various service categories by applying a design pattern. When a user selects a service, method, and method structure through the integration script generation process, a script having a desired function is generated through the combination of a script unit or a method unit. A case study of script generation via method unit and script unit combination was presented. Additionally, we showed that the integration script generation technique can support various service categories, services, and methods through comparative analysis with related studies. The script generation method allows the user to conveniently utilize various cloud functions. The case study was conducted for an integrated scenario of the selection and script generation techniques. The processes of selecting an actual cloud and generating a script were presented for a prototype. Through the proposed C-SIP, users can build their own hybrid cloud environment, which is expected to facilitate the introduction of hybrid clouds and the acquisition of cloud strategies. In future research, we plan to improve the cloud environment by analyzing the user feedback and monitoring the log of the established cloud environment.

V. COMPARISON OF VARIOUS FAULT TOLERANCE TECHNIQUES FOR SCIENTIFIC WORKFLOWS IN CLOUD COMPUTING.

Cloud Computing gives high level services that are provisioned fastly with minimum efforts frequently on Internet and it is a common pool of configurable resources of computer system. It depends on resource sharing to gain economics of scale and coherence like public utility. Third party clouds allow organisations that centres business to expand resources on computer maintenance and infrastructure. Advocates take note of that cloud computing enables organisations to keep away from or limit in advance costs of IT infrastructure. In present life everybody is dependent on internet directly or indirectly the most well-known precedent are sites such as Yahoo, Gmail, facebook that everyday gets a more number of clicks. This outcomes in the age of terabytes of significant information. So applications usually require storage, real time capturing and investigation of this information. Fault tolerance is one of the major issue. At the point when fault tolerance happens, procedures provide mechanism to the system software for avoidance of occurrence of failure system. In this paper, we have reviewed various techniques for optimising the fault tolerance for workflows that are scientific in cloud environment. The appropriate fault tolerance strategy may reduce the cost, scheduling overhead and scalability. It is impossible to satisfy all the requirements for effective fault tolerance as it violates the service level agreement. On conclusion we can say that ICFWS calculation is good, as it is able to do resubmission and replication for fault tolerance. It is assessed on both randomly generated and real world workflows. In future this technique can be build robust with elastic resource provisioning scheduling algorithm strategy for workflow in the Cloud systems.

VI. FAILURE MANAGEMENT FOR RELIABLE CLOUD COMPUTING

Based on failure management techniques and policies for reliability assurance in cloud computing, the components of the taxonomy are 1) design principle, 2) QoS, 3) architecture, 4) application type, 5) protocol, and 6) mechanism (see Figure 2). Design Principle. Three different types of design principles are proposed for reliable cloud service such as design for recoverability, i.e., the recover system with minimum involvement of human, design for data integrity, i.e., to ensure the accuracy and consistency of data during transmission, and design for resilience, i.e., the enhance system resilience and reduce the effect of failure to there is lesser interruption to cloud service.

Failure management in open source technologies: various types of open-source technologies are identified for failure management in reliability-aware approaches such as hadoop, storm, spark, kafka, zookeeper, cassandra, flink, beam, ape, and samza.

Fault-tolerance and resilience in practice: There are various commercial clouds such as Amazon Web Services, Window Azure, Google App Engine, IBM Cloud, and Oracle, which focuses on fault tolerance to deliver reliable cloud service. In this section, we have explored the recent advances of commercial cloud providers based on eight different types of fault tolerance parameters.5,6,11,13,14,17,18,22 To improve the reliability of the system, the information is shared among redundant resources (hardware or software), is called replication. The capability of a system to deliver 247 service in case of failure—a disk, a node, or a network is called availability. The capability of a system to protect against data loss during write, read, and rewrite operations on storage media is called durability. Archiving-cool storage means lower cost tier for storing data which is accessed infrequently and long-lived. Disaster recovery provides automatic replication and protection of VMs using recovery plans and its testing. Relational database provides organization of data to develop data driven websites and applications without demanding to manage infrastructure. Caching offers effective storage space, which is used to off-load nontransactional work from a database. Table 4 shows the comparison of commercial clouds based on fault tolerance parameters.

Further, the existing techniques of the reliable cloud computing have been analysed based on the taxonomy of failure management. We have discussed the failure management in open-source technologies and the fault tolerance resilience in practice for commercial clouds. Further, fault tolerance in modular micro services and the resilience on exascale systems is discussed. We propose a conceptual model for effective management of resources to improve reliability of cloud services. Moreover, a case study of astronomy workflow is presented for reliable execution in cloud environment. Our study has helped to determine research gaps in reliable cloud computing as well as identifying future research directions. Ensure the drafted journal is critically reviewed by your peers or any subject matter experts. Always try to get maximum review comments even if you are well confident about your paper.

VII. SCRUTINIZE: FAULT MONITORING FOR PREVENTING SYSTEM FAILURE IN CLOUD COMPUTING

Growing interest for the management of Cloud environment is the result of the loop holes like availability and performance. Nagios is one of the highly acceptable monitoring tool for gathering information from remote nodes so that proactive fault tolerance can be implemented[7][8].Nagios generated results and other system related data is stored in hard disk.

Table 1.Tools Comparative analysis

Tool	Monitoring Purpose	Communication medium
Nagios	Service, Application	Mail and SMS alert
Cacti	Network	Audible messages
Zabbix	Sensors	XMPP

An existing fault monitoring application Nagios core has been implemented for analyzing the fault occurrence conditions in system. This application helps in proactive fault tolerance by determining the status of the host and services running on them. Fault tolerance can be achieved only after understanding the existing faults and there root cause. By monitoring physical machine and a web application with Nagios it has become easy to understand the behavior of the system corresponding to the fault tolerance implementation. For rectification of the errors its required to identify them first and to reach to the sole reason for the occurrence of the problem or we can't handle system failures unless we are monitoring its processing time to time. A monitoring tool called Nagios is used for tracking faults in the system. Nagios is an open source tool for monitoring of physical and virtual machines It tells the current status of the hosts, devices and services which are added to its configuration files. It checks the system time to time or for specified time to detect any of the uncertain system behavior and notify the administrator or associated contact group so that preventive method can be followed for avoiding system failure. Monitoring is done by monitoring agents on the basis of metric values obtained from the monitored components. Monitoring is done in following steps:

- Collect the relevant metrics from the resources.
- Send values to the monitoring server which stores and analyze them.
- After server analysis reports are generated and notifications are sent to the contact

VIII. FINE-GRAINED FAULT TOLERANCE FOR RESILIENT PVM-BASED VIRTUAL MACHINE MONITORS

VMM is made of two components: the hypervisor and a privileged VM (pVM), The pVM acts as a control plane for the hypervisor, through a specific interface, and is involved in all VM management operations (creation, startup, suspension, migration . It also hosts I/O device drivers that are involved in all I/O operations performed by user VMs (i.e., regular VMs) on para-virtual devices. The pVM is typically based on a standard guest OS (e.g., Linux) hosting a set of control-plane daemons. This pVM-based design is popular and used in production-grade, mainstream virtualization platforms (for example, Xen, Microsoft Hyper-V and some versions of VMware ESX) for several important reasons, including the following ones: (i) it simplifies the development, debugging and customization of the control plane [1], (ii) it provides isolation boundaries to contain the impact of faults within the control plane or the I/O path [2], (iii) it offers flexibility for the choice of the OS hosting the control plane (which matters for considerations like code footprint, security features, and available drivers for physical devices) [3], (iv) it provides a data plane with a smaller attack surface than a full-blown operating system like Linux

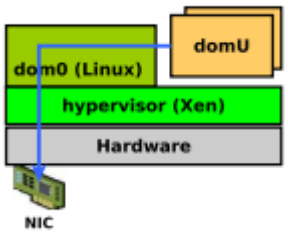

Fig 2. Overall architecture of a Xen-based virtualization platform.

VMMs remain a key building block for cloud computing, and many of them are based on a pVM-based design. We have highlighted that, in this design, the pVM component has become the main weakness in terms of fault tolerance (compared to the bare metal hypervisor component). Besides, the existing solutions only tackle a limited set of pVM services (device drivers) and/or require long failure detection/recovery times and significant performance overheads. To the best of our knowledge, our contribution is the first to propose and demonstrate empirically, a complete approach allowing to achieve both high resilience (against failures of different components and concurrent failures of interdependent services) and low overhead. Our approach currently relies on manual tuning of some important parameters (e.g., for failure detection and scheduling) but, we envision that recently published works could help manage

them in a more automated and robust way. Another area for future work is the tuning and optimization of resource allocation for disagreggrated pVM components, which could be extended from existing techniques proposed for a monolithic pVM design.

IX. FAULT-TOLERANT FEEDBACK VIRTUAL MACHINE DEPLOYMENT BASED ON USER-PERSONALIZED REQUIREMENTS.

Virtualization is an important technology that promotes the development of cloud computing. The management and service of a cloud platform can be made flexible by the virtualization of the basic resource, and the dispersedly managed software and hardware resources can be transformed into a shared resource pool with virtualized management. Users can obtain several necessary resources according to their individual demands and focus on their own business without having to address the details of the operation and maintenance of several hardware. Therefore, users can focus their energy on making business innovation and raise the utilization rate of the computing resources. The cloud platform can integrate many resources, such as computing, storage, and information service, to perform unitized management and maintenance of the resources. The characteristics of the requirements of different application programs in virtual machines to physical resources determine the specific requirements of the physical resources. The application programs of the virtual machine can be divided into computation intensive style , memory-intensive style, I/O intensive style, and network intensive style according to the requirements of the virtual machine to different resources. Moreover, virtual machine resources are correspondingly divided into four categories: CPU, memory, I/O, and network resources. The importance of each resource varies with the different application scenarios of the users. Therefore, the physical machines are distributed to the virtual machines based on the individual demands of the users, which will use each physical resource appropriately and improve the service experience of the users. However, most of the present research only consider the user's individual needs to the resources, but ignore the differences of individual demands of fault tolerance. This results in the deficiency of pertinent fault-tolerance methods in the deployment process of virtual machines and the reduction of the service experience of the user. Therefore, the deployment problem of virtual machines according to the users' resources and performance of self-adaptive fault-tolerant mechanism needs to be solved urgently.
The characteristics of the requirements of different application programs in virtual machines to physical resources determine the specific requirements of the physical resources. The application programs of the virtual machine can be divided into computation intensive style, memory-intensive style, I/O intensive style, and network intensive style according to the requirements of the virtual machine to different resources. Moreover, virtual machine resources are correspondingly divided into four categories: CPU, memory, I/O, and network resources. The importance of each resource varies with the different application scenarios of the users. Therefore, the physical machines are distributed to the virtual machines based on the individual demands of the users, which will use each physical resource appropriately and improve the service experience of the users. However, most of the present research only consider the user's individual needs to the resources, but ignore the differences of individual demands of fault tolerance. This results in the deficiency of pertinent fault-tolerance methods in the deployment process of virtual machines and the reduction of the service experience of the user. Therefore, the deployment problem of virtual machines according to the users' resources and performance of self-adaptive fault-tolerant mechanism needs to be solved urgently.
In this paper, an integrated framework of the virtual machine deployment process and the corresponding deployment scheme are introduced based on the personalized demand of the fault tolerance of users. It resolves the lack of fault tolerance or the fault tolerance cannot meet the personalized demand in the deployment process. Meanwhile, it also takes the utilization ratio of the resource into consideration in the deployment process and decreases the number of the physical node as much as possible. Combining the estimation of the user's performance according to the queuing model M/M/1 and the secondary fault-tolerance effect, it shows the quantity of the required virtual machines and physical machines under the specific condition. Thus, it can provide an effective guidance to start the physical node. The combination with the K-level redundant method can deploy the virtual machine of the same user task into different physical nodes, which resolves the failure of multiple user services caused by the physical fault of a single physical node. Under the premise that the resource of the cloud infrastructure can be fully utilized, this paper presented a deployment framework of the virtual machine based on the interaction fault tolerance of users aiming to solve the problem of high reliability and the diversity of the fault-tolerance demand of the application system. The advantage of the proposed method is primarily manifested in three layers to select the corresponding individual fault-tolerance service for users: the type of fault tolerance, the time of fault tolerance, and the level of fault tolerance. The relatively high reliability of the personalized service for users can be guaranteed through the dual-layer fault-tolerant service provided by the utilization of the underlying resource of the virtual machine and the realization of the optimized deployment of the user service. At present, the estimation and deployment of the virtual machine and the

physical node are implemented only aiming to the condition of a small number of users combined with the fault-tolerant level. In the future, we will focus on the problem where the fault-tolerant level can be adjusted in different stages of the task deployment by a large scale of users. Moreover, a deeper investigation is required on how a large-scale user node dynamically selects more reliable fault-tolerant service according to the ever-changing requirement of fault tolerance.

X. SECURITY IN HARDWARE ASSISTED VIRTUALIZATION FOR CLOUD COMPUTING— STATE OF THE ART ISSUES AND CHALLENGES

The term virtualization, in the field of computing can also refer to Container based virtualization and Application virtualization. Container based virtualization products such as the Docker, allow creation of isolated user space instances at the operating system level, without creating a complete Virtual machine. Application virtualization technology creates operating system agnostic environments for the application execution. Java Virtual Machine is a classic example of this type. The scope of our discussion is limited to the platform virtualization technologies as used in the modern day cloud infrastructures, which enable the creation and execution of a full VM capable of running its own OS. We do not consider the container based virtualization and application virtualization technologies here. We further limit our discussion on threats associated with the hardware assisted virtualization techniques and specifically on Intel's x86 and x86_64 platforms, as they continue to represent the largest CPU family used in the server segment worldwide.

Intel has introduced a number of CPU extensions and hardware features, collectively referred to as the Intel VT (Virtualization Technologies) that are aimed to simplify the development of a VMM. Intel VT-x is a set of CPU extensions aimed at supporting processor virtualization. Additional CPU instructions (called the VMX instructions) have been added as part of these extensions. Intel EPT (Extended Page Tables) helps in achieving memory virtualization. The VT-d technology aims at providing efficient I/O virtualization.

Attacks on Intel SGX Software Guard Extensions (SGX) is a new technology from Intel, for guaranteeing trusted execution of applications. This technology has been used to build trusted cloud platforms which allow applications to execute in a secure manner. SGX allows the applications to execute in isolated contexts called as enclaves, which guarantee isolation from other applications, including malicious system software which might be running with higher privileges. The technology also supports CPU based attestation, to ensure that the enclave is indeed created on an uncompromised system with trusted components. SGX gives protection to the memory pages of the applications running inside an enclave. Further, the entries and exits to the enclave are also guarded, protecting the application memory pages and register files. However as pointed out in, the threat model of SGX has excluded cache timing and side channel attacks, leaving the technology vulnerable to such attacks. The authors have explained the possibility of carrying out a powerful attack from the malicious system software targeting an application running inside the SGX enclave. Cache attacks have been demonstrated to target cryptographic implementations (including RSA and AES) running in SGX enclaves. SGX does not disable hyper-threading in the processor. This gives rise to the possibility of creating side channels by attackers for stealing information from other processes running on the same CPU. A tracing kernel or a malicious hypervisor can also be used to create memory address translation attacks on applications executing inside a SGX enclave, specifically to generate a page level trace of the enclaved application execution. Page fault attacks have also been demonstrated to target cryptographic implementations executing under SGX. Page fault side channel in SGX has also been demonstrated byBulck et al. have also built a Linux kernel framework for carrying out SGX attacks through controlled execution inside the enclave. In, SGX has been shown to be vulnerable to shared data synchronization bugs. It has been demonstrated that these bugs can be exploited to arbitrarily affect the enclave thread scheduling process.

The power of virtualization technologies and the advent of cloud computing, indicate that these technologies are here to stay. Nevertheless, their adoption cannot accelerate further, unless the concerns on security are well addressed. We have thus presented a detailed survey of the topics and challenges pertaining to security in hardware assisted virtualization. We have also discussed the possible countermeasures and open challenges that remain in these technologies. An in-depth treatment and enumeration of the issues and known attacks at different components of the virtualization stack, will help the researchers to understand the breadth of issues related to the security aspects of using hardware virtualization. The survey can also educate the users of public clouds. Research efforts from multiple system domains are required to tackle the security issues arising at various virtualization components. We hope the challenges mentioned here would enthuse the researchers to work collaboratively and find effective defenses against the known threats to virtualized environments.

X. MULTI LEVEL FAULT TOLERANCE IN CLOUD ENVIRONMENT

Cloud computing provides different kind of services to users. With the increased demand of services, chances of fault are also high. To reduce the impact of the fault, many fault tolerance techniques have been designed and proposed. This paper proposed a multilevel fault tolerance system to tolerate the faults in real time cloud environment by providing higher reliability and availability. In the first level, reliable virtual machines are identified using reliability assessment algorithm and in the second level availability of data is achieved through replication mechanism. Thus the system provides an efficient fault tolerant mechanism in multi-level fashion

Fault tolerance is the ability of cloud computing nodes to operate continuously even in the presence of faults. It is the quick replacement and repairing of faulty nodes in case of failure. These faults may arise due to hardware failure, virtual machine malfunctioning, network congestion and applications failure. Fault Management in a cloud computing environment depends on two major parameters. ¾ Recovery point objective: It defines the volume of data lost during fault. ¾ Recovery Time Objective:

In the proposed work the fault tolerance mechanism is achieved in two levels. In the first level, the cloud users store their real time application in the buffer. From the buffer, all the applications are fed to multiple virtual machines for processing. After processing all the application in the virtual machine, the results are forwarded to the acceptance test. The acceptance test verifies whether all the virtual machine produces correct logical output or not. Then, the acceptance test forwards its results (pass or fail criteria) to the time checker. The time checker checks that all the virtual machine performs its operation within the limit or not. If the virtual machine produces the result on time, it gives the result to the reliability assessor. The reliability assessor assesses the reliability of the virtual machine by using the reliability assessor algorithm. Initially all the virtual machine reliability is set as 100%. If the virtual machine gives its result on time, its reliability increases otherwise the reliability of the virtual machine decreases. After assessing all the virtual machine reliability in the reliability assessor it gives the result to the decision making algorithm. In the decision making algorithm it takes only the best reliable virtual machine among all the virtual machines.

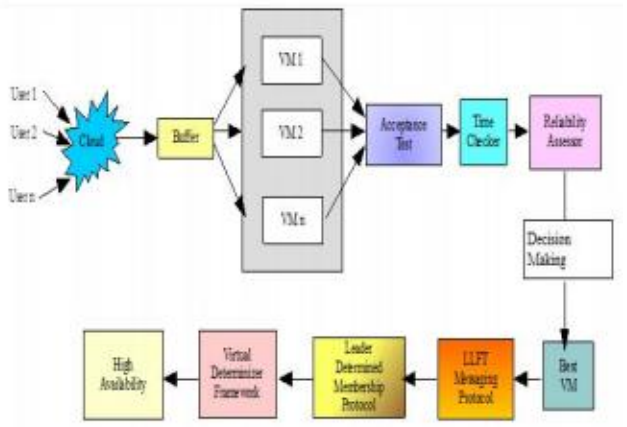

Fig 3.Architecture of Multi level fault tolerance.

Tolerating the fault in the cloud environment is one most important factor in mission critical applications. Many existing systems have addressed the fault tolerant systems in single level. The paper proposes a multilevel fault tolerance system in the cloud. In the first level, the best reliable VMs are identified using reliability assessment algorithm and in the second level replication is provided using LLFL and the Leader determined membership protocol. Thus the system tolerates the faults in an efficient way by proving high reliability and high availability.

XII. FAULT TOLERANCE MANAGEMENT IN CLOUD COMPUTING: A SYSTEM-LEVEL PERSPECTIVE

The increasing popularity of Cloud computing as an attractive alternative to classic information processing systems has increased the importance of its correct and continuous operation even in the presence of faulty components. In this paper, we introduce an innovative, system-level, modular perspective on creating and managing fault tolerance in Clouds. We propose a comprehensive high-level approach to shading the implementation details of the fault tolerance techniques to application developers and users by means of a dedicated service layer. In particular, the service layer allows the user to specify and apply the desired level of fault tolerance, and does not require knowledge about the fault tolerance techniques that are available in the envisioned Cloud and their implementations.

The task of offering fault tolerance as a service requires the service provider to realize generic fault tolerance mechanisms such that the client's applications deployed in virtual machine instances can transparently obtain fault tolerance properties. To this aim, we define ft−unit as the fundamental module that applies a coherent fault tolerance mechanism to a recurrent system failure at the granularity of a VM instance. The notion of ft−unit is based on the observation

that the impact of hardware failures on client's applications can be handled by applying fault tolerance mechanisms directly at the virtualization layer than the application itself (e.g., [8], [9]). For instance, fault tolerance of the banking service can be increased by replicating the entire VM instance in which its application tier is deployed on multiple physical nodes, and server crashes can be detected using well-known failure detection algorithms such as the heartbeat protocol. An example of a heartbeat protocol is depicted in Fig. 2, where the primary and backup components are run in VM instances independent of the banking service's application tier. In this example, the primary component periodically sends a liveness request to all backup components and maintains a timer for each request. When a backup receives a liveness request, it immediately responds to the primary. If the backup fails (due to a server crash) to respond to the primary for N consecutive requests, each within a predefined timeout threshold, it is suspected to failure. In this context, we note that replication of the client's application (ft−unit1), and detection of node failures (ft−unit2) are performed without requiring any changes to the application's source code. In this paper, we assume that the service provider realizes a range of fault tolerance mechanisms as ft−units, and based on this assumption we present a two-stage delivery scheme: design stage, and runtime stage, to transparently deliver high levels of fault tolerance to client's applications using ft−units.

s. In particular, we presented an approach for realizing generic fault tolerance mechanisms as independent modules, validating fault tolerance properties of each mechanism, and matching user's requirements with available fault tolerance modules to obtain a comprehensive solution with desired properties. The proposed approach when combined with our delivery scheme enables a service provider to offer long-standing fault tolerance support to client's applications. Furthermore, we designed a framework that allows the service provider to integrate its system with the existing Cloud infrastructure and provides the basis to generically realize our approach in delivering fault tolerance as a service. The components of our framework can be extended to improve the overall resilience of the Cloud infrastructure. Our future work will mainly be driven toward the implementation of the framework to measure the strength of fault tolerance service and to make an in-depth analysis of the cost benefits among all the stakeholders. R

XIII.CONCLUSION

Cloud computing demand is increasing due to which it is important to provide correct services in the presence of faults also. The Resources in cloud computing can be dynamically scaled that too in a cost effective manner. Fault Tolerance is the process of finding faults and failures in a system. If a fault occurs or there is a hardware failure or software failure then also the system should work properly. Failures should be managed in a effective way for reliable Cloud Computing. It will also ensure availaibility and robustness .This paper aims to provide a better understanding of fault tolerance techniques which are used for managing faults in cloud. It also deals with some existing Fault tolerance model. Tolerance of faults makes an important problem in the scope of environments of cloud computing. Fault tolerance method activates when a fault enters the boundaries i.e theoretically these strategies are implemented for detecting the failures and make an appropriate action before failures are about to occur.

REFERENCES

[1] Vinay K, Kumar SD, Raghavendra S, Venugopal KR. Cost and fault-tolerant aware resource management for scientific workflows using hybrid instances on clouds. Multimedia Tools and Applications. 2018 Apr 1;77(8):10171-93.

[2] SuruchiTalwani, Jimmy Singla Computer Science and Engineering Lovely Professional University, Phagwara, India suruchitalwani14@gmail.com, jimmy.21733@lpu.co.in "Comparison of Various Fault Tolerance Techniques for Scientific Workflows in Cloud Computing" in 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), India, 14th -16th Feb 2019.

[3] Anu Computer Science Department, Thapar University, Patiala, India Anju Bala, "Enhancing Fault Tolerance of Cloud Nodes using Replication Techniques" in International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020

[4] International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020

[5] MeriemAzaiez SOIE-COSMOS Laboratory National School of Computer Science University of ManoubaManouba, Khaled Ghedira SOIE-COSMOS Laboratory National School of Computer Science University of ManoubaManouba, "Hybrid Fault Tolerance Model for Cloud Dependability" in 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems.

[6] I. P. Egwutuoha, D. Levy, B. Selic, and S. Chen, "A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance

computing systems," The Journal of Supercomputing, vol. 65, no. 3, pp. 1302–1326, 2013.

[7] Joonseok Park &Ungsoo Kim &Donggyu Yun &KeunhyukYeom "Approach for Selecting and Integrating Cloud Services to Construct Hybrid Cloud" in J Grid Computing https://doi.org/10.1007/s10723-020-09519-x,

[8] K.Devi Department of Computer science and Engineering Valliammai Engineering College Kattankulathur, Chennai devik.cse@valliammai.co.in Dr.D.Paulraj Department of Computer Science and Engineering R.M.D Engineering College Chennai kingrajpaul@gmail.com "Multi Level Fault Tolerance in Cloud Environment" in International Conference on Intelligent Computing and Control Systems ICICCS 2017

[9] Seyyed Mansur Hosseini and Mostafa GhobaeiArani "Fault-Tolerance Techniques in Cloud Storage: A Survey" International Journal of Database Theory and Application Vol.8, No.4(2015),pp.183-190

[10] DjobMvondo∗ Alain Tchana† Renaud Lachaize∗ Daniel Hagimont‡ Noël De Palma∗∗ Univ. Grenoble Alpes, "Fine-Grained Fault Tolerance For Resilient pVM-based Virtual Machine Monitors" in 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)

[11] MiloudKhaldi, Mohammed Rebbah, BoudjelalMeftah& Omar Smail "Fault tolerance for a scientific workflow system in a Cloud computing environment" in ISSN: 1206-212X (Print) 1925-7074 (Online) Journal homepage: https://www.tandfonline.com/loi/tjca20

[12] Manimaran G, Murthy CSR. A fault-tolerant dynamic scheduling algorithm for multiprocessor real-time systems and its analysis. IEEE Trans Parallel Distrib Syst. 1998;9(11):1137–1152.

[13] Zhu X, Wang J, Guo H, et al. Fault-Tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized Clouds. IEEE Trans Parallel Distrib Syst. 2016;27(12):3501–3517.

[14] B. Asvijaa,∗ , R. Eswari b , M.B. Bijoya "Security in hardware assisted virtualization for cloud computing—State of the art issues and challenges" in https://doi.org/10.1016/j.comnet.2019.01.013 1389-1286/© 2019 Elsevier B.V. All rights reserved.

[15] Sukhpal Singh Gill and Rajkumar Buyya Cloud Computing and Distributed Systems Laboratory, School of Computing and Information Systems, The University of Melbourne "Failure Management for Reliable Cloud Computing: A Taxonomy, Model, and Future Directions" in April 30,2020 at 02:26:13 UTC from IEEE Xplore.

[16] R. Jhawar and V. Piuri, "Fault tolerance and resilience in cloud computing environments," in Computer and Information Security Handbook. 3rd ed., 2017, pp. 165–181.

[17] R. Jhawar and V. Piuri, "Fault tolerance and resilience in cloud computing environments," in Computer and Information Security Handbook. 3rd ed., 2017, pp. 165–181.

[18] H. Casanova, F. Vivien, and D. Zaidouni, "Using replication for resilience on exascale systems," in Fault-Tolerance Techniques for High-Performance Computing. Cham, Germany: Springer, 2015.

[19] D.P. Chandrashekar, "Robust and fault-tolerant scheduling for scientific workflows in cloud computing environments," Ph.D. Thesis, The Univ. Melbourne, Parkville, VIC, Australia, Aug. 2015.

[20] Anu Computer Science Department, Thapar University, Patiala, India Anju Bala "Fault Monitoring for Preventing System Failure in Cloud Computing" in International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 4, Special Issue May 2015

[21] Shukun LIU1, Weijia JIA2, Xianmin PAN "Fault-tolerant feedback virtual machine deployment based on user-personalized requirements"inFront.Comput.Sci. https://doi.org/10.1007/s11704-017-6422-0

[22] Zhou A, Wang S, Cheng B, Zheng Z, Yang F, Chang R, Buyya R. Cloud service reliability enhancement via virtual machine placement optimization. IEEE Transactions on Services Computing, 2017, 10(6): 902–913