Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan Vol. 22 No. 4-2017
ISSN:1813-1786 (Print) 2313-7770 (Online)

Fault Tolerance Techniques in Cloud and Distributed Computing- A Review

M. Hussain[1], S. Farid[2]

[1,2]*Computer Science Department, Bahauddin Zakariya University, Multan, Pakistan.*
[2]shahidfarid@bzu.edu.pk

Abstract-Computing System is a combination of different computers and associated software which can share common memory. In cloud and distributed computing the defective infrastructure is the main issue. Fault Tolerance (FT) techniques minimize the failure effect on computing environment. This study discusses and presents a comprehensive analysis of the current FT techniques in cloud and distributed environment. A comparison and critical evaluation of common FT techniques in cloud and distributed computing is also performed. An intensive literature review was adopted to identify current FT techniques in both environments. This study contributes in a fashion by identifying that few fault tolerance techniques are common in both cloud and distributed computing .Moreover this study also pinpoints Check-pointing as the most optimal FT technique. Result shows that FT techniques are crucial for reliability and availability of data thus FT method is necessarily required to be endorsed to escort the system from failures.

Keywords-Fault Tolerance, Cloud Computing, Distributed Computing, Replication, High Availability, Reliability.

I. INTRODUCTION

Cloud computing is the result of increasing demand of flexibility in computing resources and storage technologies in cost-effective manner. Cloud computing provides reliable services which include storage, servers and networks. A cloud system is developed by connecting computing resources and extensive data centers provided over the internet as on-demand service with the help of virtual machines. Distributed Computing System (DCS) is a single coherent system of independent computers. A DCS connect computers at different locations using local networks[i].Although cloud and distributed computing has been extensively used in the industry, still there is a need to completely report several research issues such as fault tolerance, workflow management, workflow scheduling and security etc. [ii].One of the crucial issues is Fault Tolerance (FT) which permit a system to endure faults pertaining in the system after development.

FT computing means if hardware/software failures occur the job can be done correctly. FT abilities are necessary to overcome the effect of system failures[iii]. One of the major benefits of applying fault tolerance in cloud and distributed computing may include improving reliability, recovery from failure, 24/7 availability, lower cost, improved performance metrics etc. [iv]. However, for application to be installed in cloud or distributed system, a complete fault tolerance solution is difficult to design that can combine the failure impact and system architecture. In distributed computing failures or faults are limited and the hardware/software redundancy approaches are famous methods of fault tolerance. Hardware techniques assure the accumulation of hardware components like memory, I/O devices, communication media and CPUs. Software FT techniques include special programs to protect system from faults. A well-organized FT tool assists to identify and recover from faults.

The remainder of the paper is planned as follows: Section II describes literature review. In Section III fault tolerance techniques and types in cloud and distributed computing are deliberated. Section IV gives a comparison of different fault tolerance techniques in cloud and distributed computing. Critical evaluation about the techniques is performed in Section V and the last but not least Section VI expose the limitations of the study whereas Section VII delineates the conclusion and explore the future directions.

II. LITERATURE REVIEW

With the passage of time various researchers came up with different FT techniques that are common with cloud and distributed computing.

FT techniques and challenges in cloud environment are discussed in[ii], [v]. An autonomic FT system and cloud virtualized system architecture has been proposed in [ii] using HA Proxy and My SQL to implement proposed FT system which can deal with several software faults for server applications. Whereas some of FT algorithms which share workload between resources such as multiple computers or a computer

cluster, central processing units, disk drives or network links are identified in [v] to attain optimal resource utilization, minimize response time, maximize throughput and avoid overload. A FT framework for High Performance Computing (HPC) in cloud is presented in [vi-vii]. A FT framework presents in [6] is implemented on Linux which perform parallel job, event logging, environmental, and resource monitoring to analyze reliability of HPC system. Framework base on proactive FT method to avoid failures and is able to gather and analyze data and cause migration. A Framework is proposed in [vii] using Process Level Redundancy (PLR), FT policy and live migration. The proposed framework reduces the execution time and ensures that Message Passing Interface (MPI) implemented with computational intensive applications run smoothly. According to [viii] an innovative, modular and system level perspective is proposed to create and manage FT in Cloud systems. A comprehensive high-level approach propose to highlight the FT techniques' implementation details to users and application developers through a dedicated service layer. In distributed environments the Low Latency FT (LLFT) middleware deployed within a cloud and data center environment provided in [ix] by means of the leader/follower replication method. The LLFT middleware retains strong replica consistency, achieves low end-to-end latency and offers application transparency. A brief overview is presented in [x] for the need to perform FT in cloud computing. An outline of the prevalent architectures and the existing techniques for FT in cloud computing has been analyzed and compared. The primary concepts of FT techniques used according to the policies like Reactive and Proactive, and the related FT tools used on various types of faults are highlighted in [xi], [xii]. A comprehensive taxonomy of faults, errors and failures is also presented in [xi]. The practice of taxonomy and survey identify the similarities as well as the areas needing for future research. Whereas [xii] discusses and implements various fault tolerant methods, frameworks and algorithms which will assist to build a robust FT technique in Cloud environment. Authors of [xiii] critically analyze the Integrated Virtualized Fail over Strategy (IVFS) model and present a new model which tolerate faults of each virtual machine based on reliability. The results show an increase in pass rates and use diverse software tools to deliberate forward/backward recovery. Simulation results of this research highlight a noble performance as compared to the existing models. Experimental study with a critical analysis is used for validation, laying the foundation for a FT IaaS Cloud system. A survey based on FT implementation tools is conducted in [xiv] about the crucial FT technique in cloud computing, and list various FT methods. Cloud Computing is a fresh field of research than other technologies particularly in developing a standalone FT method. Several FT methods of this field are proposed by the research experts. The document [iii] presents a proactive FT framework which can implement several proactive FT techniques such as pause/unpause and migration. As well as allow the implementation of new proactive FT policies. Presented work implement and experiment proactive FT on system-level virtualization and results are compared with simulation results. Understanding of FT mechanisms have been proposed in [xv] which are used to manage faults in clouds and deals with existing FT model. Now a days a number of proposed models provide mechanisms to improve the system performance. But still there is a need to concern about every frame work. The objective of [xvi] is to propose a fault tolerant model and fault detection algorithm to overcome the holes of previously applied algorithms using Artificial Neural Network, provided in cloud computing. To detect faults in cloud systems artificial Neural Network is used which first detect the faults and then appropriate proactive FT technique such as pre-emptive migration or check-pointing apply to keep the system fault tolerant. Exhaustive study has been presented in [xvii] on FT in cloud, and introduce FT types and limitations. Furthermore, this study represents how much dependable the FT system will be for the real time environments. Results shows that an autonomic FT technique is needed to handle faults, when one of the machines stop working while several instances of an application are running on multiple machines.

The article [xviii] emphasized the different FT techniques for multiple system failures prevention by considering high availability, high redundancy and replication in distributed computing. The levels of FT are also addressed such as the hardware FT ensures additional backup hardware and software FT systems includes rollback recovery mechanisms and checkpoints storage. A survey on several FT techniques and issues in distributed environment has been done in [xix]. The survey results help to explore the future guidelines about FT mechanism. Important issues about FT tools such as adaptive, confidence, completeness, accuracy, fastness, and able to detect several faults. Faults, failures and errors categorization has been presented briefly in [xx] which are encountered in a distributed system. Furthermore, different techniques for fault tolerance and identification offered in Grid and Clustered Computing systems has been examined. It is also proposed that a standard FT framework should be proficient to handle all the recognized faults, failures and errors. Fault identification and foretelling in the view of Clusters is concentrated in [xxi] and efforts are done to develop a method that predicts the faults based on temperature in

clustering systems. A comparison of fault detection and FT techniques in distributed environment is also performed. Various techniques for FT in distributed computing systems are provided in [xxii]. Every technique has some pros and cons. Check pointing can be used to improve performance and makes system fault tolerant during the case of node mobility and node failure. Various approaches for FT are discussed in [xxiii] and concluded that FT keeps system dependable and more reliable. Two stages in FT are the fault detection and fault recovery. Furthermore it is concluded that FT improves performance to achieve reliability. Centralized logging process described in [xxiv] by using message logging for message losses. Proposed logging process has three central parts failure detection, overload detection and failure recovery. Proposed system sustains monitor nodes in cluster for all nodes. All monitor nodes can perform task as a cluster node at the time of node failure as well as also during the system functioning properly. Concurrent replication with canceling are explored in [xxv] and a stochastic model is proposed to study replication affects in the cluster Service Level Objectives (SLOs). The proposed model is reliable as well as defines the regions of the utilization. Additionally, the model can support resource provisioning decisions with response time guarantees and reliability. Several issues associated to check pointing in distributed and mobile computing systems are discussed in [xxvi]. For distributed systems a survey is performed about some check pointing algorithms. Results of survey support check pointing technique as an effective FT technique for this reason it require minimum storage and avoids the domino effect. Nowadays, storage and manipulation of huge amount of data is done by cloud computing environment. For the reason availability and reliability is the major concern for manipulation of data. Different FT mechanisms and techniques to manage resources, available in cloud infrastructure are discussed by [xxvii]. The comparison of two state-of-the-art FT techniques is performed by [xxviii]. The comparison is done in terms of energy cost to cloud employers and availability of services to customers. The study concluded that proactive FT policy redundancy is remarkable in terms of cost to cloud manipulators and provides availability and services to customers. Cloud computing architecture (servers, networks), key concepts and application components are highlighted in [xxix]. Furthermore a qualitative overview of failures that occur in cloud infrastructure is also highlighted. Moreover cloud computing architecture resiliency techniques are categorized in the proposed study. Efforts to inspect FT techniques are analyzed in [xxx]. A taxonomy of faults and FT aspects are presented in the study. Moreover, different failure models, tools, metrics and support systems are classified. Weaknesses and strengths of current FT techniques are also discussed. In today's world cloud computing is a developing field but still faces some major problems. Such as fault tolerance, resource discovery, security and load balancing. A detailed summary of techniques to optimize load balancing is provided by [xxxi]. Swarm based and evolutionary algorithms are discussed to reduce the resource utilization and optimization problems. A new model is proposed with results in [xxxii] to provides FT for cloud framework. The proposed model is able to tolerate the faults by making the adaptive decision. The decision is made by utilizing the resource allocation of the jobs with a new technique in real time cloud locality.

Literature review illustrates that previous work on fault tolerance in cloud and distributed environment is about fault tolerance techniques, their uses, surveys and analysis. In this study a comprehensive analysis of various FT techniques in cloud and distributed computing have been done. Additionally this study also contributes in a way by comparing different FT techniques which have been commonly proposed for both cloud and distributed computing. Furthermore we critically evaluate common FT techniques in cloud and distributed computing.

III. Fault Tolerance

Fault Tolerance (FT) considered as a setup or configuration that precludes a network device or a computer system from getting failed due to any system failure or error in the system and consider effective steps to prevent from failure [viii]. FT quickly repair and replace the faulty devices to keep the system in working state. A fault tolerant system is associated to dependable system. Dependability contains few valuable requirements such as Reliability, Availability, Safety and Maintainability in a fault tolerant system: [xviii].

Availability: Availability concerned with working systems in time at given instant. System should be in ready state to facilitate users.

Reliability: Ability of a system to do job uninterruptedly with no chance of failure occurrence. Reliability concerned with time interval as an extremely reliable system works constantly without interruption for long time period.

Safety: Safety concerned with system failure when its operations are incorrect and cannot complete its processes correctly.

Maintainability: Maintainability of a system shows excessive measurement of accessibility when the system failure can be detected and fixed automatically.

A. Types of Fault Tolerance

Fault tolerance is categorized in two categories [x]. Hardware FT can be attained by implementing extra

hardware such as communication links, processors and other resources. Software fault tolerance deals with fault messages when added into the system.

1) Hardware Fault Tolerance:

Hardware FT provide delivery of supplementary hardware backup like Memory, CPU, Power Supply Units and Hard disks. Hardware fault tolerance cannot deal with accidental interfering and errors within software programs. This technique have focused towards structuring systems that can recover themselves from the faults, and involves splitting a computing system into modules which can backed up with a self-protective redundancy to continue its function if failure occurs.

2) Software Fault Tolerance:

Software FT also apply dynamic or static redundancy approaches like hardware fault tolerance. Rollback recovery and checkpoints storage are software FT methods. The effectiveness of software FT is to develop an application to keep checkpoints repeatedly for required system.

B. Fault Tolerance Techniques

For real-time applications two types of Fault Tolerant (FT) polices are available.

1) Reactive Fault Tolerance Policy

Reactive approaches decrease the fault effects by taking essential actions. Some fault treatment policies can also be used to prevent faults from being reactivated.

2) Proactive Fault Tolerance Policy

Proactive approaches predict errors, faults and failures and replace the suspected components.

C. Fault Tolerance Techniques in Cloud Computing

Usually, one of the pillars of software reliability is fault avoidance. Since cloud architecture is very complex and built on cloud data centers that consist of multiple interconnected servers. Hence considering fault prevention techniques in developing stage is quite monotonous. Fault avoidance techniques help to detect and remove fault but not sufficient to achieve reliability, so fault tolerant system is necessary. FT in cloud computing is the capability to resist with changes occurred due to network congestions and hardware/software failures. Several factors to classify cloud computing faults are as follows: [xxxiii] xxvii, [xxxiv]

Network fault: Network faults occur due to the reason of packet corruption, destination failure, network partition, link failure, and packet loss etc.

Process faults: Software bugs and shortage of resources are reasons behind process fault.

Processor faults: When operating system crashes processor fault occurs.

Physical faults: Hardware fault are physical faults such as storage, memory and CPU failure etc.

Service expiry fault: Service expiry faults occurs if during processing a resource service time expires.

Media faults: Media head crashes and some other media factors cause's media faults.

Fault tolerance techniques in cloud environment are classified into various types[xi] as shown in Fig. 1.

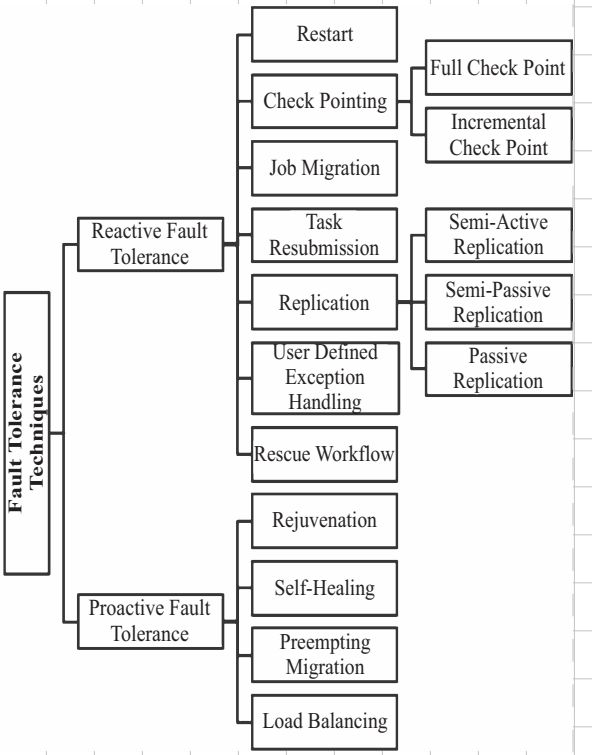


Fig. 1. Cloud Computing Fault Tolerance Techniques

1) Reactive Fault Tolerance Technique

Failure of application execution can be reduced by Reactive FT. It helps in recovery of system state from an unstable state to stable state so that the system can again start working to provide desired results. The major techniques under this category are as follows:

i) Restart

Restart functions on application level or on programs. A task is suspected to have aborted, failed and restarted, if not completed within a given time period[xxxv].

ii) Check pointing

In Check pointing system's state save in regular or irregular time intervals. Check pointing is done on every change in a system. At whatever time a job failure occurs, recent checked point state is used to restart the job. Check pointing technique is additionally categorized as follows.

- *Full Checkpoint*

Checkpoints are applied to a running process after a fixed time interval and the process state save on some media. If process failure occurs during execution then last saved checkpoint state is used to recover.

- *Incremental Checkpoint*

This mechanism helps in reducing the checkpoint overhead by saving those pages in which there have been any change instead of saving the whole process

iii) Replication Replication means producing copies of similar data and run them on multiple resources. Replication can be further classified as follows.

- *Semi-Active Replication*

Each replica is provided by input or state information. The primary and backup replica accomplish execution on the provided input and main replica produces the output. If the main replica goes down, backup replica produces the output. VM ware's FTis an example of semi-active replication group.

- *Semi-Passive Replication*

State information is moved to all the backup replicas in semi-passive replication mechanism. Main replica save input parameters between checkpoints. Backup replica save the newest state gained by primary replica. When primary replica fails backup replica starts and is updated as primary replica. Example of semi-passive replication is Remus.

- *Passive Replication*

Virtual machine instance state information is stored regularly as a backup. If failure occurs, FT manager restores the last saved state by recommission another Virtual Machine (VM) instance. The backup can share the state of some VM instances or can be used for a specific application. Example of passive replication is VM ware's High Availability solution.

iv) Job Migration

HA-Proxy is used to migratea task to another machine during failure.

v) Task Resubmission

The failed task is resubmitted to a new or same resource when failure of task occurred.

vi) User Define Exception Handling

User's predefined actions are used whenever fault is detected. Such as user define the behavior of a task when failure occurs.

vii) Rescue Workflow

Workflow is an order of connected steps that complete execution without gap or delay, earlier then successive stage may commence [v]. When a task fails, rescue workflow allows the task to continue until or unless it is not possible to execute without considering the task which failed.

2. Proactive Fault Tolerance Technique

Proactive FT procedures helps to neglect faults by foretelling before faults occur and replacing doubted constituents with other operational constituents before they actually occur. The major techniques are as follows:

i) Rejuvenation

Rejuvenation is issued before the system fails. The system is designed for sporadic reboots. Whenever any failure occurs the system is restarted with new clean and fresh start.

ii) Self-Healing

Self-healing method based on divide and conquer technique, several parts of an application runs on different VMs and failures are handled automatically.

iii) Preemptive Migration

In preemptive migration executing job is based on feedback loop control technique. Before migration system save the current state of job and then transferred to other system.

iv) Load Balancing

When memory and CPU load exceeds a specific limit, the load is migrated to other CPU.

Also, there are mainly some other techniques that can be used for fault tolerance.

These are:

- *Safety bag checks*

In safety bag checks the commands which does not fulfill the safety requirements are blocked.

- *Retry*

In order to complete the task on one virtual machine only, retry method is used and the task is implemented repeatedly on the same resource.

- *S-Guard*

S-Guard is based on rollback recovery and is less tempestuous to regular stream processing.

- *Alternate resource*

This technique allows to find an alternate resource for current virtual machine instead of retrying the same resource or changing virtual machine instance.

D. Fault Tolerance Techniques in Distributed Computing

Real time distributed systems are highly responsible on deadline such as robotics, grid and nuclear air traffic control systems. A system can fail if any mistake in real time distributed system is not accurately detected and recovered in time. Fault tolerance is a way to keep the system in working condition during failure. In distributing computing

system hardware and software infrastructure delivers dependable, consistence and inexpensive high end computations[xxxvi]. Many FT techniques available in distributed environment are represented in Fig. 2.

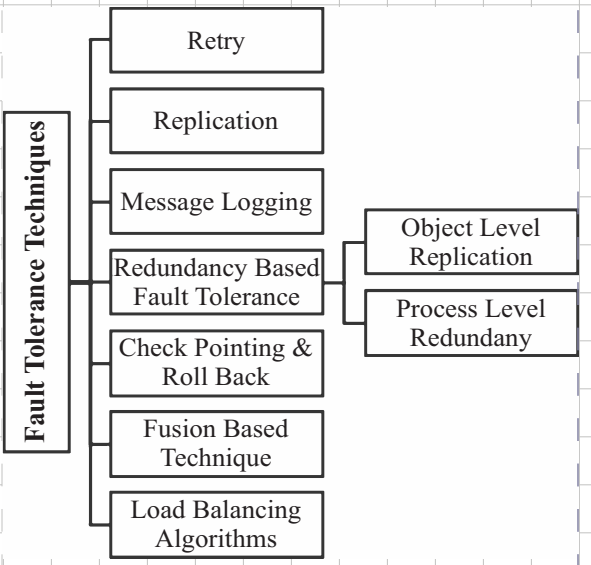


Fig. 2. Cloud Computing Fault Tolerance Techniques

i) Retry

The ordinary failure recovery mechanism in which the outcome will not be affected in succeeding repeats without considering the cause of failure[37].

ii) Replication

Replication is the method of creating and maintaining various copies of data objects and processes. In replication based procedure duplicate copies of a task runs on various machines until all replicated tasks are not crashed[xxxvii].

iii) Message Logging

In message logging technique all contributing nodes log received messages to constant memory so that a consistent global state is computed when a failure encountered. Algorithms of this methodology can be categorized into optimistic and pessimistic message logging[xxxviii].

iv) Redundancy based fault tolerance

Redundancy is having more than one functionally ready components of a system other than a component that actually provides the service. Process level and data or object level are the two levels in which we can implement the redundancy. This approach uses replication based technique to create redundancy[xxii].

- *Object Level Replication Based Technique*

Replication is making several copies of related data on the servers [xxxix]. In replication based technique, client request is promoted from a collection of replicas to one of replica. Object level replication is proxy based monitoring technique which has two approaches; Active or Passive and supports to improve performance of the system and remove complexity and overhead. It is a well-known method used to improve the availability. Replication increases redundancy and inconsistency in system. In this technique system will not fail if some nodes will fail and hence fault tolerance is accomplished as shown below in Fig. 3. [xviii].

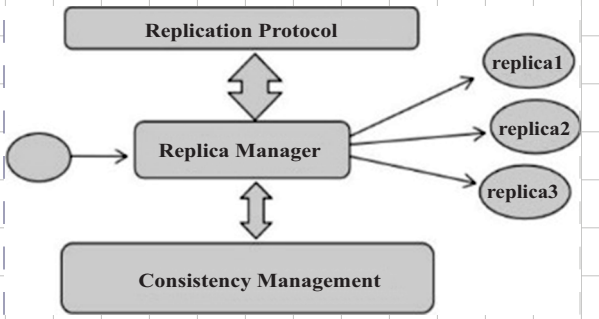


Fig. 3. Object Level Replication Based Technique

- *Process Level Redundancy*

Process Level Redundancy (PLR) is a method of creating duplicate processes for each application process as shown in Fig. 4 [xl]. PLR is transient fault tolerance which is software based technique. Transient faults are difficult to handle and identify and they are evolving as a serious matter for reliability of distributed environment. PLR analytically compares the processes to assure correct execution and permits the operating system to program the processes with presented hardware resources. PLR offers enhanced performance with 16.9 percent over current transient FT mechanism with additional fault detection overhead[xli].

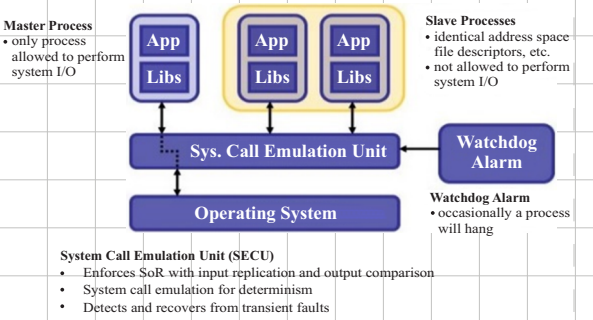
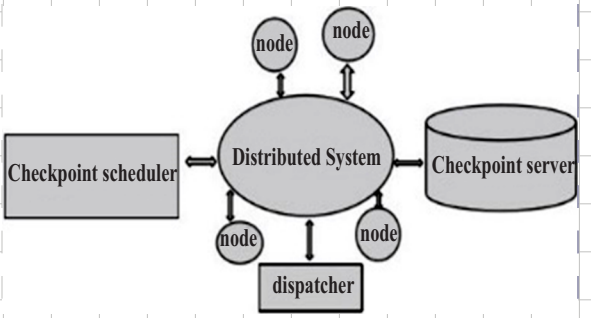


Fig. 4. Process Level Redundancy

ISSN:1813-1786 (Print) 2313-7770 (Online)

v. Check Pointing and Roll Back

Checkpoint is more famous method to restore the process after failure at certain points. This technique saves the present stage of computation in constant memory to be used when node failure occurs. Checkpoints are periodically recognized during the execution of program. The checked or stored information contains the process state and register's value. The process rolls back to last saved state when error is detected. Fig. 5. gives an idea about this method[xviii].



vi. Fusion Based Technique

Fusion based techniques overcome the problem of backups created in replication method as managements of multiple backups is so much expensive. Fusion based technique is evolving as a famous procedure to deal with various faults as it requires a few backup machines which can be managed easily. Back up machines (fusion corresponding machine) [xlii] are cross product of original machines. Overhead increases during recovery from faults in fusion based technique. Fig. 6 represents the basic idea of fusion based technique[xxii].

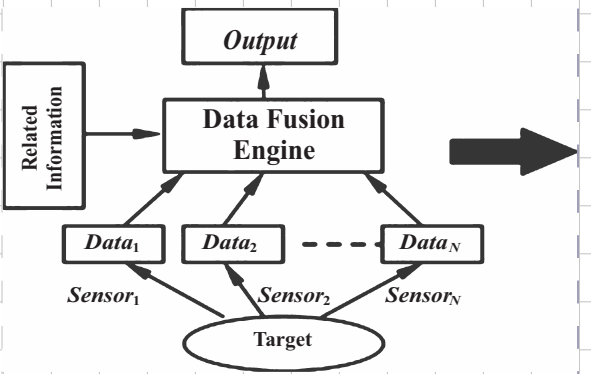


Fig. 6. Fusion Based Technique

vii. Load Balancing Algorithms

During execution time load balancing algorithms reallocate the processes among the processors. The algorithm improves the performance of the system by allocating task from heavy weighted task to light weighted task. Dynamic load balancing schemes has some drawback such as[xliii]:

- Runtime overhead; since the information of load is transported between processors.
- The transmission delay; since the job relocate to itself.
- Decision making process; the choice of processors and processes for transfer of job.

IV. Comparison

Choice and evaluation of a fault tolerance technique is based on some factors. These factors are consistency management, multiple faults handling, working procedures efficiency, multiple failure detection and appropriate performance for a system is the major task. The performance can enhanced by working on the critical issues discussed in this section on the basis of comparison performed in Table III.

A. Comparison of FT Techniques in Cloud Computing

Comparison of FT techniques in cloud computing is illustrated in Table I. A detailed comparison of both Reactive and Proactive polices is highlighted. Comparison is performed on the basis of tools, programming framework, environment used, type of fault and application type. Furthermore key features are also discussed in order to elaborate the comparison between techniques. Therefore on the basis of comparison as shown in Table I, it can be safely concluded that FT techniques in cloud are reliable and have the capability of detecting and handling multiple faults.

TABLE I
COMPARISON OF FT TECHNIQUES IN CLOUD COMPUTING

FT Techniques	Polices	Tools used	Programming Framework	Environment	Type of Fault Detected	Application Type	Key Features
Replication	Reactive	HA-Proxy, Hadoop, AmazonEC2	JAVA, Amazon Map, Amazon Machine Image	Virtual Machine	Node failure, Process failure	Load Balancing, Fault Tolerance	• The task runs on various VM Instances at the same time. • More resource utilization.
Check Pointing	Reactive	Assure, SHelp	JAVA, SQL	Virtual Machine	Application Failure	Fault Tolerance	• Effective for long running applications. • Provides efficient resource utilization
Retry	Reactive	Assure	JAVA	Virtual Machine	Host failure, Network failure	Fault Tolerance	• Job is retried on the same resource. • Time inefficient.
Job Migration	Reactive	Hadoop, HA-Proxy	JAVA, HTML, CSS	Cloud Environment	Application failure, Node failure, Process failure	Data intensive	• If job cannot be executed on same machine then it can be migrated to different machine. • More resource utilization • Time efficient
Task Resubmission	Reactive	AmazonEC2	Amazon Machine Image, Amazon Map	Cloud Environment	Application failure, Node failure	Load balancing, Fault Tolerance	• Job is retried on same or different resource. • Time inefficient and more resource utilization.
S-Guard	Reactive	Hadoop, AmazonEC2	JAVA, HTML, CSS	Cloud Environment	Application failure, Node failure	Data intensive	• Less tempestuous to ordinary stream processing. • Centered upon rollback recovery.
Self-Healing	Proactive	HA-Proxy, Assure	JAVA	Virtual Machine	Host failure, Network failure	Load balancing, Fault Tolerance	• Larger task is divided into subtasks. • Good overall performance.

A. Comparison of Fault Tolerance Techniques in Distributed Computing

Comparison of FT techniques in distributed computing is illustrated in Table II. A detailed comparison of Reactive FT techniques is highlighted. Comparison is performed on the basis of major factors such as working, multiple faults handling, performance, N-faults, multiple failure detector and consistency management. Techniques used for comparison are replication based technique, check pointing, roll back, fusion based techniques and process level redundancy.

TABLE II
COMPARISON OF FT TECHNIQUES IN DISTRIBUTED COMPUTING

Major Factors	Fusion Based Technique	Checking Pointing and Roll Back	Replication Based Technique	Process Level Redundancy
Working	Machine Back up	State saved for recovery on stable storage	Forwarded to replica	A collection of duplicate process
Multiple Faults Handling	Affected by number of back up machines	Affected by Check pointing scheduling	Affected by degree of replica	Affected by set of duplicate processes
Performance	Decrease due to faults as high recovery cost	Decrease by size and frequency of checkpoint	Decreases when number of replicas increases	Decrease when faults appears disappear
N-Faults	N backups machine required to handle extra N faults	Uncoordinated, Pessimistic and N level disk less used for N-1 Faults	N replicas ensure n-1 faults	Scaling the number of process and majority voting
Multiple Failure Detector	Reliable, Accurate and Adaptive.	Reliable, Accurate and Adaptive.	Reliable, Accurate and Adaptive.	Reliable, Accurate and Adaptive.
Consistency management	Has to be implemented among backup machines	Avoiding orphan messages	Strategies like active or passive replication	Can be easily implemented

C. Comparison of Common FT Techniques in Cloud and Distributed Computing

Comparison of FT techniques in both cloud and distributed computing is illustrated in Table III. A detailed comparison of Reactive FT techniques and common in both types of computing is highlighted. Comparison is performed on the basis of crucial features and type of fault detected. Comparison is done on retry, replication, check pointing, job migration and task resubmission.

TABLE I
COMPARISON BETWEEN CLOUD AND DISTRIBUTED COMPUTING FT TECHNIQUES

FT Techniques	Cloud Computing		Distributed Computing	
	Type of fault detected	Key Features	Type of fault detected	Key Features
Retry	• Host failure • Network failure	• Job is retried on the same resource. • Time inefficient	• Task Crash failure	• Cannot handle user defined exceptions
Replication	• Process • Node Failure	• The task runs on various VM Instances at the same time. • More resource utilization.	• Host crash • Network failure	• Exploiting task's stateless and idempotent nature • Enhance performance and availability. • Multiple copies available • Inconsistency • Expensive
Check Pointing	• Application failure	• Effective for long running applications. • Provides efficient resource utilization	• Application failure	• Bigger application will take more time • Consistency. • Identification of user in some cases such as user triggered check pointing.
Job Migration	• Application failure • Node failure • Process failure	• If job cannot be executed on same machine then it can be migrated to different machine. • More resource utilization • Time efficient	• Host crash • Network failure • K crash	• Does not support diverse failure recovery mechanism
Task Resubmission	• Application failure • Node failure	• Job is retried on same or different resource. • Time inefficient • More resource utilization.	• Host failure • Network failure	• Cannot handle user defined exceptions

D. Discussion

It is found through intensive literature review that two basic polices of FT are proactive and reactive FT. However reactive FT policy is considered only for comparison of cloud and distributed computing. Some reactive fault tolerance techniques are common in both type of computing such as Retry, Replication, Check pointing, Job migration and task resubmission.

Retry in cloud computing is time inefficient and host and network failure are type of faults detected. Retry in distributed computing cannot handle user defined exceptions and task crash failure is a type of fault detected. When Replication is used, type of fault detected in cloud computing is process and node failure and in distributed computing is host crash and network failure. In Cloud computing replication can run task on various VM instances at a time and more resources are utilized. In distributed computing replication exploits task stateless and in idempotent nature. It enhances performance and availability but is inconsistent and expensive. Check Pointing is a technique in which application failure occur in both type of computing as a type of fault. Check pointing is effective for long running applications and provides efficient resource utilization when used in cloud computing and is consistent when used with cloud computing as well. Check pointing may provide identification of user in some cases. Application, node and process failure occur in cloud computing and host crash, network failure and K crash occur in distributed computing in case of Job Migration. Job migration is time efficient but utilizes more resources and job can be migrated to other machine in cloud computing. It does not support diverse failure recovery mechanism in distributed system. In cloud computing application and node failure may occur. Task resubmission is done which is time inefficient and it requires more resources as well as job is retried on same or different resource. In distributed computing task resubmission cannot handle user defined exception and host and network failure are types of faults that are detected.

V. CRITICAL EVALUATION

A comparison of common fault tolerance techniques in cloud and distributed computing is performed and critically evaluated as follows:

- *Cloud Computing*

The cloud systems still cannot provide there liability, robustness and quality required for the processing of numerous work flows[ii]. Fault tolerance techniques are complex and inter-dependent which need careful analysis and consideration. An autonomic fault tolerance mechanism is needed by using various constraints in cloud systems. Through the intensive literature review and comparative analysis several limitations of fault tolerance integration of cloud environment are stated below;

Different genres of systems in the cloud environment is the major difficulty to focus on the faults. During job migration job is transferred to different machines, if it cannot be executed on same machine. So an efficient fault tolerance technique is needed. FT technique will not be effective if different VMs run multiple instance of a job[xxxiv]. Replication is a technique which divides task on various machine

and causes more resource utilization. When processing is completed on remote computers, error chances are increased. When task resubmission is used job is retried on same or different machine which is inefficient and utilizes more resources. Interpretation of changing system state is difficult since cloud environments are dynamically scalable, provide virtualized resources as a service. Check pointing will be difficult to save system state in regular and irregular time intervals.Host and network failure is the major issue of retry technique of cloud computing based on network connection. If network breaks down whole communication of data will stop.

- *Distributed Computing*

Taking checkpoint in distributed environment is a difficult task as any random set of checkpoints cannot be used for recovery. Since the checkpoints used for recovery must form a reliable global state. Replication enhances redundancy in a system. Duplicate copies give rise to inconsistency as various users may update diverse files. A huge number of replicas is required for high level of consistency since low quantity of replicas affect the performance and scalability. The main disadvantage is duplication of backups since the backups increase and the management cost upsurges to a great extent when faults are higher. Drawback of using retry or task resubmission technique is that it cannot handle user defined exceptions. Job migration does not support diverse failure recovery mechanism.

IV. LIMITATIONS

The study is limited to focus on reactive FT techniques only. A detailed study on proactive fault tolerance techniques is required to be done to pinpoint the most effective and potent technique.

V. CONCLUSION

Cloud and distributed computing has become influential computational technology over the last few years. Reliability and availability of the systems are of the most important requirements. Since a competent FT method is necessary to guards the system from failures or faults. FT system is one of the foremost parts of any system as it guarantees the system to continue its working during fault or failure. Two major components of fault tolerance are failure detection and recovery. In this study different stages of fault tolerance are highlighted. Hardware fault tolerance guarantees the extra backup of hardware such as CPU and memory. Software FT ascertain rollback recovery mechanisms and checkpoint storage. Every technique has some pros and cons. Some techniques are advantageous while the others are costly. In this study some of the fault tolerance techniques that are common in both cloud and distributed computing have been identified and critically evaluated.

It is concluded that reactive FT policy check-pointing is the optimal among all the techniques in both type of computing. The study also affirms that proactive FT policy redundancy is remarkable in terms of availability. In case of node agility or node failure check pointing improve performance and makes system fault tolerant. Furthermore, it depicts that there is necessity of a more effectual and reliable technique which is cheaper than the prevailing techniques. In all these techniques, accurate, reliable, and pure adaptive multiple failure detector mechanism is needed. To achieve reliability, future work will primarily be motivated to evaluate the benefits of FT services over a cloud and distributed environment.

REFERENCE

[i] T. Dimovski and P. Mitrevski. "*Connection Fault-Tolerant Model for distributed transaction processing in mobile computing environment in Information Technology Interfaces (ITI),*" Proceedings of the ITI 2011 33rd International Conference 2011. IEEE.

[ii] A. Bala and I. Chana, "*Fault tolerance-challenges, techniques and implementation in cloud computing.*"IJCSI International Journal of Computer Science Issues, 2012. 9(1): p. 1694-0814.

[iii] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia. "*Above the clouds: A berkeley view of cloud computing*, Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley. 2009.

[iv] G. Vallee, K. Charoenpornwattana, C. Engelmann, A. Tikotekar, C. Leangsuksun, T. Naughton and S. L. Scott. "*A framework for proactive fault tolerance.*" Third International Conference on Availability, Reliability and Security, 2008. ARES 08. 2008. IEEE.

[v] Y. M. Essa, "*A Survey of Cloud Computing Fault Tolerance: Techniques and Implementation.*" International Journal of Computer Applications, 2016. 138(13).

[vi] A. Litvinova, C. Engelmann, and S. L. Scott. "*A proactive fault tolerance framework for high-performance computing.*"Proceedings of the 9th IASTED International Conference. 2009.

[vii] I. P. Egwutuoha, S. Chen and D. L. B. Selic, "*A fault tolerance framework for high performance computing in cloud.*" Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012). 2012. IEEE Computer Society.

[viii] R. Jhawar, V. Piuri and M. Santambrogio, "*Fault tolerance management in cloud computing: A system-level perspective.*" IEEE Systems

Journal, 2013. 7(2): p. 288-297.

[ix] W. Zhao, P. M. Smith, and L. E. Moser, "*Fault tolerance middleware for cloud computing*." IEEE 3rd International Conference on Cloud Computing (CLOUD). 2010. IEEE.

[x] P. D. Kaur and K. Priya, "*Fault tolerance techniques and architectures in cloud computing-a comparative analysis*." International Conference on Green Computing and Internet of Things (ICGCIoT). 2015. IEEE.

[xi] H. Agarwal and A. Sharma, "*A comprehensive survey of Fault Tolerance techniques in Cloud Computing*." 2015 International Conference on Computing and Network Communications (CoCoNet). 2015. IEEE.

[xii] A. Ganesh, M. Sandhya and S. Shankar, "*A study on fault tolerance methods in cloud computing*." Advance Computing Conference (IACC), IEEE 2014.

[xiii] B. Mohammed, M. Kiran, I. U. Awan and K. Maiyama, "*An Integrated Virtualized Strategy for Fault Tolerance in Cloud Computing Environment*." Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC /ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences. 2016. IEEE.

[xiv] S. M. Ataallah, S. M. Nassar and E. E. Hemayed. "*Fault tolerance in cloud computing-survey*." 11th International Computer Engineering Conference (ICENCO), 2015. 2015. IEEE.

[xv] D. Mittal, and N. Agarwal, "*A review paper on Fault Tolerance in Cloud Computing*." Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference. 2015. IEEE.

[xvi] Z. Amin, H. Singh, and N. Sethi, "*Review on fault tolerance techniques in cloud computing*." International Journal of Computer Applications, 2015. 116(18).

[xvii] P. Gupta and S. Banga, "*Topic-Review of Cloud Computing in Fault Tolerant Environment With Efficient Energy Consumption*."IJSRM 2013.

[xviii] Sari, A. and M. Akkaya, "*Fault Tolerance Mechanisms in Distributed Systems*." International Journal of Communications, Network and System Sciences, 2015. 8(12): p. 471.

[xix] S. Bansal, S. Sharma, and I. Trivedi, "*A detailed review of fault-tolerance techniques in Distributed System*." International Journal on Internet and Distributed Computing Systems, 2011. 1(1): p. 33.

[xx] S. Haider, N. R. Ansari, M. Akbar, M. R. Perwez and K. M. U. Ghori, "*Fault tolerance in distributed paradigms*." International Conference on Computer Communication and Management, Proc. of CSIT. 2011.

[xxi] S. Haider and N. R. Ansari, "*Temperature based fault forecasting in computer clusters*." 15th Multitopic International Conference (INMIC), 2012. IEEE.

[xxii] P. Kaur and K. Mahajan, "*Various Techniques for Fault Tolerance in Distributed Computing System- A Review*." International Journal of Computer Science and Mobile Computing, 2015. 4(5, May 2015): p. 754 – 759

[xxiii] A. Shwethashree, D. V. Swathi, "*A Brief Review Of Approaches For Fault Tolerance In Distributed Systems*." International Research Journal of Engineering and Technology (IRJET) 2017. 4(2-Feb-2017).

[xxiv] A. Patil, A. Shah, S. Gaikwad, A. A. Mishra, S. S. Kohli and S. Dhage, "*Fault Tolerance in Cluster Computing System*." International Conference *on* P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2011. 2011. IEEE.

[xxv] Z. Qiu, and J. F. Pérez, "*Enhancing reliability and response times via replication in computing clusters*." *IEEE Conference on Computer Communications (INFOCOM), 2015*. 2015. IEEE.

[xxvi] R. Garg, and P. Kumar, "*A Review of Checkpointing Fault Tolerance Techniques in Distributed Mobile Systems*." International Journal on Computer Science and Engineering, 2010. 2(04): p. 1052-1063.

[xxvii]S. Dhasarathapandian, K. Dhanalakshmi and S. Somasundaram, "*A survey on various fault tolerant and resource management techniques in cloud computing framework*." Advances in Natural and Applied Sciences, 2017. 11(7): p. 301-306.

[xxviii]A. M. Sampaio and J. G. Barbosa, "*A Comparative Cost Study of Fault-Tolerant Techniques for Availability on the Cloud*." *International Symposium on Ambient Intelligence*. 2017. Springer.

[xxix] C. C. Meixner, C. Develder, M. Tornatore and B. Mukherjee, "*A survey on resiliency techniques in cloud computing infrastructures and applications*. IEEE Communications Surveys & Tutorials, 2016. 18(3): p. 2244-2281.

[xxx] D. Poola, M. A. Salehi, K. Ramamohanarao and R. Buyya, "*A Taxonomy and Survey of Fault-Tolerant Workflow Management Systems in Cloud and Distributed Computing Environments*." Software Architectures for Cloud and Big Data Book, 2016.

[xxxi] A. Dave, B. Patel and G. Bhatt, "*Load balancing in cloud computing using optimization techniques: A study*." International Conference on Communication and Electronics Systems

(ICCES), 2016. IEEE.

[xxxii]P. K. Patra, H. Singh, R. Singh, S. Das, N. Dey, A. D. C. Victoria, "*Replication and Resubmission Based Adaptive Decision for Fault Tolerance in Real Time Cloud Computing: A New Approach*." International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), 2016. 7(2): p. 46-60.

[xxxiii]L. M. Vaquero, L. R. Merino, J. Caceres and M. Linder, "*A break in the clouds: towards a cloud definition*. ACM SIGCOMM Computer Communication Review, 2008. 39(1): p. 50-55.

[xxxiv]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*." IEEE International Conference on Cluster Computing (CLUSTER), 2010. IEEE.

[xxxv]K. Wolter, "*Stochastic Models for Fault Tolerance: Restart, Rejuvenation and Checkpointing*." Springer Science & Business Media, 2010.

[xxxvi]A. Costan, C. Dobre, F. Pop, C. Leordeanu and V. Cristea, "*A fault tolerance approach for distributed systems using monitoring based replication*. IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2010. 2010. IEEE.

[xxxvii]S. Hwang and C. Kesselman, "*A flexible framework for fault tolerance in the grid*." Journal of Grid Computing, 2003. 1(3): p. 251-272.

[xxxviii]A. P. Sistla and J. L. Welch, "*Efficient distributed recovery using message logging*." Proceedings of the eighth annual ACM Symposium on Principles of distributed computing. 1989. ACM.

[xxxix]M. Wiesmann, F. Pedone, A. Schiper, B. Kemme and G. Alonso, "*Understanding replication in databases and distributed systems*." 20th International Conference on Distributed Computing Systems, 2000. IEEE.

[xl] A. Shye, T. Moseley, V. J. Reddi, J. Blomstedt and D. A. Connors, "*Using process-level redundancy to exploit multiple cores for transient fault tolerance*." 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2007. DSN'07. 2007. IEEE.

[xli] A. Shye, J. Blomstedt, T. Moseley, V. J. Reddi and D. A. Connors, "*PLR: A software approach to transient fault tolerance for multicore architectures*. IEEE Transactions on Dependable and Secure Computing, 2009. **6**(2): p. 135-148.

[xlii] V. K. Garg, "*Implementing Fault-Tolerant Services Using State Machines: Beyond Replication*."DISC. 2010. Springer.

[xliii] R. Singh and M. Dave, "*Using host criticalities for fault tolerance in mobile agent systems*."2nd IEEE International Conference onParallel Distributed and Grid Computing (PDGC), 2012 2012. IEEE.