STUDY ON FAULT TOLERANCE AND CHECKPOINT IN CLOUD ENVIRONMENT

[1]Sridepa T, Research Scholar, PG & Research Department of Computer Science, Government Arts College (Autonomous), Karur – 639005, Tamilnadu, India
[2]Dr. V. Baby Deepa, Assistant Professor, PG and Research Department of Computer Science, Government Arts College (Autonomous), Karur – 639005, Tamilnadu, India

Abstract: Cloud computing is an emerging platform for compose computing and storage space reachable to the end-users as services. The computing resources like software, etc and services can be competently delivered and utilized, making the vision of computing usefulness attainable. To make sure effective performance, identification and fault tolerance should be taken into account. Cloud Computing of Fault Tolerance (FTCC) is a significant part of research due to its difficulty. Still, there is requiring of studies in these fields and it is the main anxiety to assurance dependability and accessibility of serious services in addition to the execution of the application. Failures must be proactively and estimated handled in turn to reduce the failure collision on the execution of the application and the system. Before failures really occur, forecast these failures and take a suitable action by using the fault tolerance method. To enhance fault tolerance capability of cloud by different fault detection techniques and architectural models have been anticipated. The major focus is fault identification, fault tolerance (fault rectification) by using the checkpoint approaches. In popular cloud providers express the need for more effectual results to recent failures and ease of use issues. In this paper, aims to give a better sympathetic of Fault Tolerance, in the cloud computing environment, different types of fault detection and tolerance techniques are used.

Keywords: Cloud Computing, Resources, check point techniques, Fault tolerance

I. INTRODUCTION

Large scale computing environment such as the clouds propose to offer access to a huge set of heterogeneous resources. A cloud is a distributed and parallel structure consisting of a grouping of virtualized and interconnected computers that are dynamically reachable as one or more unified computing resources [1]. The shared software, resources, and information provided through the cloud to computer and other device are normally offered as metered services in excess of the Internet. For the User, cloud computing provide numerous service. In the cloud computing, now days the user level is extremely increased to make use of the services. Cloud services are delivered either as storage or computing services. Google, iCloud and Drop box are well known examples of storage services further, Amazon EC2 and Microsoft Azure are the best examples for computing services. Though Cloud Computing broadly used everywhere but still it comes with lot of fault related issues. Fault or failure can occur anywhere anytime that may lead to loss of data, business loss, financial loss and customers trust loss too. Fault can of many reasons like due to some Human error like fault in design or imperfect design, bugs in application or some security breaches (Hacking, DDOS attacks etc). The fault appears as resources failure, such as storage/applications, hardware that is being used by the end users in a representative cloud environment. There are a variety of faults which can take place in cloud computing, like Hardware Failure, Virtual Machine (VM) Failure and Application Failure [2]. The faults may be: **(a)** software bug and design error, **(b)** intermittent, transient or permanent hardware fault, (**c)** externally induced fault and error and **(d)** operator error. To avoid all such failure a system must be designed that is fault tolerant in nature. System can be declared as fault tolerant that is able to produce optimal result even after the failure occurs. There are two techniques used to make system fault tolerant i.e. Proactive and Reactive fault tolerance techniques. Proactive techniques are those that used to avoid any upcoming failure through prediction but reactive techniques are those techniques that are used after the fault occurs. In cloud environment, the fault tolerance technique facilitates robustness and dependability. Implementation of the fault tolerance technique most important advantages [3] in cloud computing is: improved performance criteria, low cost, failure recovery, and so on. The key inspiration of the study is find out a variety of accessible cloud computing fault tolerance models and techniques is to sustain researcher to add in mounting additional efficient algorithm. In this paper, planned in cloud computing to deliberates about a variety of faults aspect and necessitate of the fault tolerance. The paper is prepared as follow. In Section 2 present the conception of cloud computing. The study of fault tolerance presented in section 3 and various FT techniques are done in Section 4. Section 5 presents the classification of checkpoint based fault tolerance techniques. Finally, the checkpoint challenges is presented in section 6 and conclusion is presented in Section 7.

II. CLOUD COMPUTING

Cloud computing is the network based upon computing wherever virtual shared servers provided infrastructure, software, devices, platform, and other customers do not own the material infrastructure somewhat they rent out the custom from a third party provider.

They use resources as a service for performing a task and pay only for what they are utilizing. Huge amount of data is stored in many cloud servers and the collection of servers forms a Data center. There are numerous benefits in cloud computing. Major benefits are, Cost Saving, Time Saving, Scalability and Flexibility, Backup and Recovery, Resource Maximization, Mobile Access, Multisharing, Customization, Collaboration, Deliver new services. Cloud computing can be organized in one of at least four special configurations: a public cloud, a community cloud, a private cloud, and a hybrid cloud. Public Cloud, in this deployment model services and infrastructures are made available to different types of customers and used publicly by general people/users. Private Cloud, in this type of cloud the computing resources are used and operated exclusively by one organization owing that cloud. Community Cloud, in community model the infrastructure is shared by a number of organizations with the same compliance and policy considerations. Hybrid Cloud, This deployment models aid the business to take benefit of data hosting and secured applications on private cloud, whereas still having the cost benefits. Cloud computing is capable to offer a multiplicity of services at the moment but major three services are Software-As-A-Services, Infrastructure-As-A-Services and Platform-As-A-Services also called as service model of Cloud computing [5].

The core computing resources are hardware and software components. They arrange the fundamentals of every computing infrastructure. Infrastructure-as-a-Structure service of cloud computing provides these services to cloud end users. SaaS is the top layer of cloud computing services. It is different than traditional software services, where usual software need own software and hardware components, Where SaaS makes users, sovereign of their own resources. Users use the integrated services provided by cloud operator. One of the best examples of SaaS is Google Docs [6, 7]. Platform as a service provides a development platform to its users so that they can develop and maintain their applications and cloud specific utilities. It is unusual from SaaS since SaaS is the developed application and deployed application and PaaS offers a ground or to platform develop those applications. Some Examples of Cloud Service Providers are, Google, Google provide services like: - Google docs, G-mail, Picasa, Google Ad words, Ad sense and Google Analytics. Microsoft, services provided by Microsoft are: - Windows Azure, SQL Azure, and Windows Azure App Fabric and Windows Azure Marketplace. Amazon Web Services (AWS), services provided AWS are:- Amazon Simple Storage services, Amazon Relational Database, Amazon Virtual Private Cloud, Amazon Simple Queue services, Amazon Elastic cloud compute, and Amazon Cloud front.

Cloud Computing is one of the most dominant field of computing resources online because sharing and management of resources is easy using cloud. These properties have prepared it an active component in the subsequent fields as follows: E-Learning, Enterprise resource planning (ERP), E-Governance, Research areas, Agricultural, and so on. Although in industry cloud computing has been widely adopted and has many benefits as mentioned above, as per fully addressed still various research issues are workflow scheduling, fault tolerance, security, workflow management, and so on.

III. FAULT TOLERANCE

In cloud computing, fault tolerance is one that can persist to properly perform its task in the failure occurrence of software failures and/or hardware or to operate adequately in the faults occurrence. To facilitates the dependability and robustness, by which fault tolerances bear-on with every inevitably technique. Fault tolerance key benefits consist of lower cost, failure recovery, improved performance metrics in the cloud computing. Dependability is associated to some Quality of Services(QoS) characteristic provided by the system, it consist of the attributes like availability and reliability. Robustness guides to the property for providing of a correct service in a difficult situation arising due to a tentative system environment. In cloud platform contains three layers: applications, virtual machines and hardware. Every one of them has a malfunction. During the program execution, the failures can be present on any virtual machine layer or hardware. Hence, appropriate action should be carrying out due to this nature of failure. The faults be able to split up in various categories [8, 9] in cloud computing.

3.1 Types of Faults

The faults can be classified based on a number of factors.

- **Physical faults:** This Faults which occurs in hardware similar to the Fault in storage, memory, CPUs, Power failure and so on.
- **Network faults:** When the data is not destined at that this fault arises for a variety of reasons such as closed date, packet loss, link failure, destination failure, and so on. These faults as regards to the networks are called as network faults.
- **Processor faults:** Due to operating system failure these faults come about on the processor.
- **Media faults:** Due to the lack of communication media the mistakes that arise.
- **Service termination fault:** The application still requires the use of resources, when the service life of the resource is over.
- **Process faults:** Due to the inefficient processing capabilities, software bugs, shortage of resource, and so on this fault which arises.
- **Intermittent failure:** This failure ensues occasionally. Whereas the system performs its operations this failures occur. Due to these failures, the damage caused although it's very complex to be identified.
- **Transient:** These types of fault very last for a lengthy time and only once appear during the actions and after the actions are completed it were disappear. Example, at first the network message cannot reach its destination from the origin; however later than a while it reach its destination prosperity.
- **Stable:** These manners of failure still survive in the system after defective systems are replaced or repaired in some cases completely.

3.2 Fault Tolerance Types

The fault tolerance is majorly categorized into two types i.e. software and hardware fault tolerance. Flout Tolerance of hardware can be attained by implementing extra hardware such as communication links, processors and other resources. Software fault tolerance deals with fault messages when further add into the system.

- **Hardware Fault Tolerance:** Hardware FT provides 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. Fault masking and dynamic recovery are the common hardware fault tolerance approaches.
- **Software Fault Tolerance:** Software FT also applies 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.

IV. FAULT TOLERANCE TECHNIQUES

There are a number of techniques use to implement the FTCC (Fault Tolerance in Cloud Computing). As shown in Figure 1 there are two main categories or policies in FT techniques i.e. Proactive and reactive Fault Tolerance. Through recovery techniques the reactive fault tolerance techniques are effort to maintain the service. After it has raised the faults, reactive fault tolerances remove the faults. While the failures happen, it does reduce the consequences of the failures on the system. By alternating with working components [10], proactive fault tolerances calculate the possible incident of faults in progress and avert the failures.

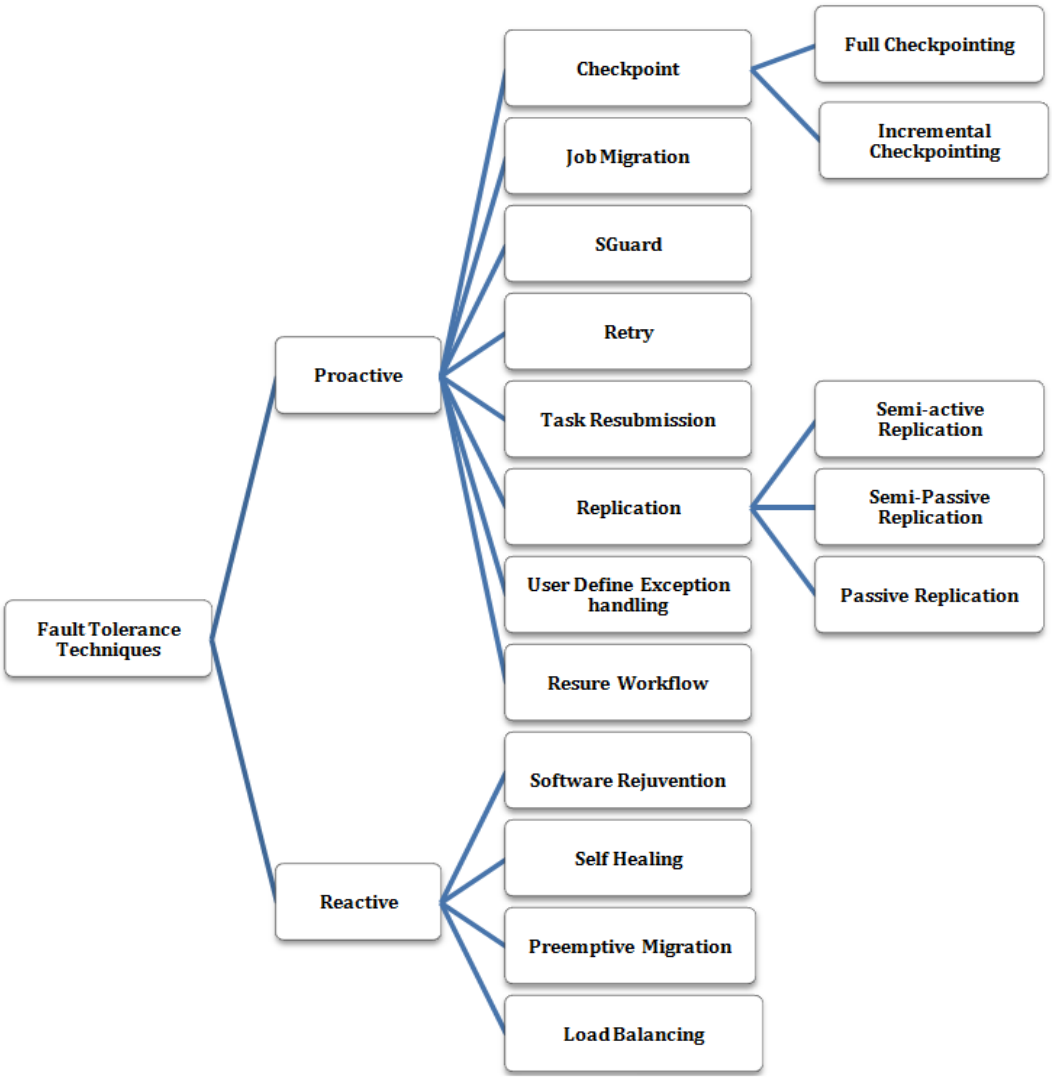

Figure 1: Fault Tolerance techniques

A. Proactive fault tolerance

The Proactive fault tolerance principle is to fault and failures proactively substitute the suspected component, expect the fault and avoid the recovery from errors. Before it actually comes it detects the problem. From crashing running parallel applications by preventatively transferring parts of an application (process, task, or virtual machine) absent from nodes that are regarding to fail, it turns away the compute node failures. Few techniques based upon these policies are using self-healing [11], Software Rejuvenation, and Preemptive migration.

- Job Migration: Due to a few reason a job cannot be entirely executed on a particular machine sometimes it happens. Task tin can migrated to a different machine, at the time of crash of whichever task. Job migration is able to implement by using HA-Proxy.
- **Replication:** It is one, in which majority important fault-tolerant technique in storage space centers that broadly used in the online service systems and in the laboratory settings. Different tasks are copied for doing well execution and best possible results, so replication performs on the various resources. Replication meaning 'to copy'. Replication is able to be executing through Hadoop, AmazonEC2 and HA-Proxy.

i. **Semi-Active Replication:** Each replica is provided by input or state information. The primary and backup replica accomplishes 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.

ii. **Semi-Passive Replication:** State information is moved to the entire backup replicas in semi-passive replication mechanism. Main replica saves input parameters between checkpoints. Backup replica saves 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.

iii. **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.

- **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.

  i. **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.

  ii. **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

- **S-Guard:** It is fewer unstable to the normal stream process. S-Guard is stand on the rollback recovery. The S-Guard is able to implement in Amazon EC2, HADOOP.
- Rescue workflow: These techniques [13] permit the workflow to persist still if the task be unsuccessful until it becomes not possible without catering the failed task to shift forward.
- **Retry:** This techniques deal with the transient type of faults. When fault is detected a retry mechanism is applied to recover from the effect of fault. This activated mechanism makes the defective module retry its activity for a certain time period. If fault remain longer than the retry period then it will be considered as permanent fault and faulty node will be replaced. If fault disappears in between retry period then it will be considered as transient fault and system will start normal functioning after recovering from it. Retry period must belong sufficient to make the transient fault disappear and short adequate to avoid overlapping of faults.[12]
- **Exception handling of User-defined:** To describes the exact action of task failure by the user for the workflows.
- **Task Resubmission:** The job may be fail now. In this case at a runtime re-submitted the task to the some other machine on which it was operating or to same machine.

B. RFT(Reactive Fault Tolerance)

These strategies while failure effectively takes place then reduce the outcome of failures on the execution of the application and it's called as ondemand fault tolerance or reactive fault tolerance. A variety of techniques based on this strategies like Replay-retry, Checkpoint/Restart, task resubmission, user-defined exception handling, recue workflow, retry, job migration, S-Guard, etc. [14-18]

- **Software Rejuvenation:** Proposes the method for the periodic reboots by this technique. Software rejuvenation restarts (starts again) the system with clean state and assist to fresh start.
- **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.
- **Self-Healing:** Huge task are separated into parts. For the better performance, multiplication is completed. It automatically handles application instances failure while different application illustrations are running on different VM.
- **Load Balancing**: During execution time load balancing algorithms reallocate the processes among the processors. The algorithm improves performance of the system by allocating task from heavy weighted task to light weighted task. When memory and CPU load exceeds a specific limit, the load is migrated to other CPU.

V. CHECKPOINT BASED FAULT TOLERANCE

Checkpoint is a process of taking a snap of the current state of running application on to a stable storage space. Checkpoint based fault tolerance is the majority frequently used method. The application can be start again from the latest checkpoint state while a fault is encounter. These decrease the re-computation time measurably.

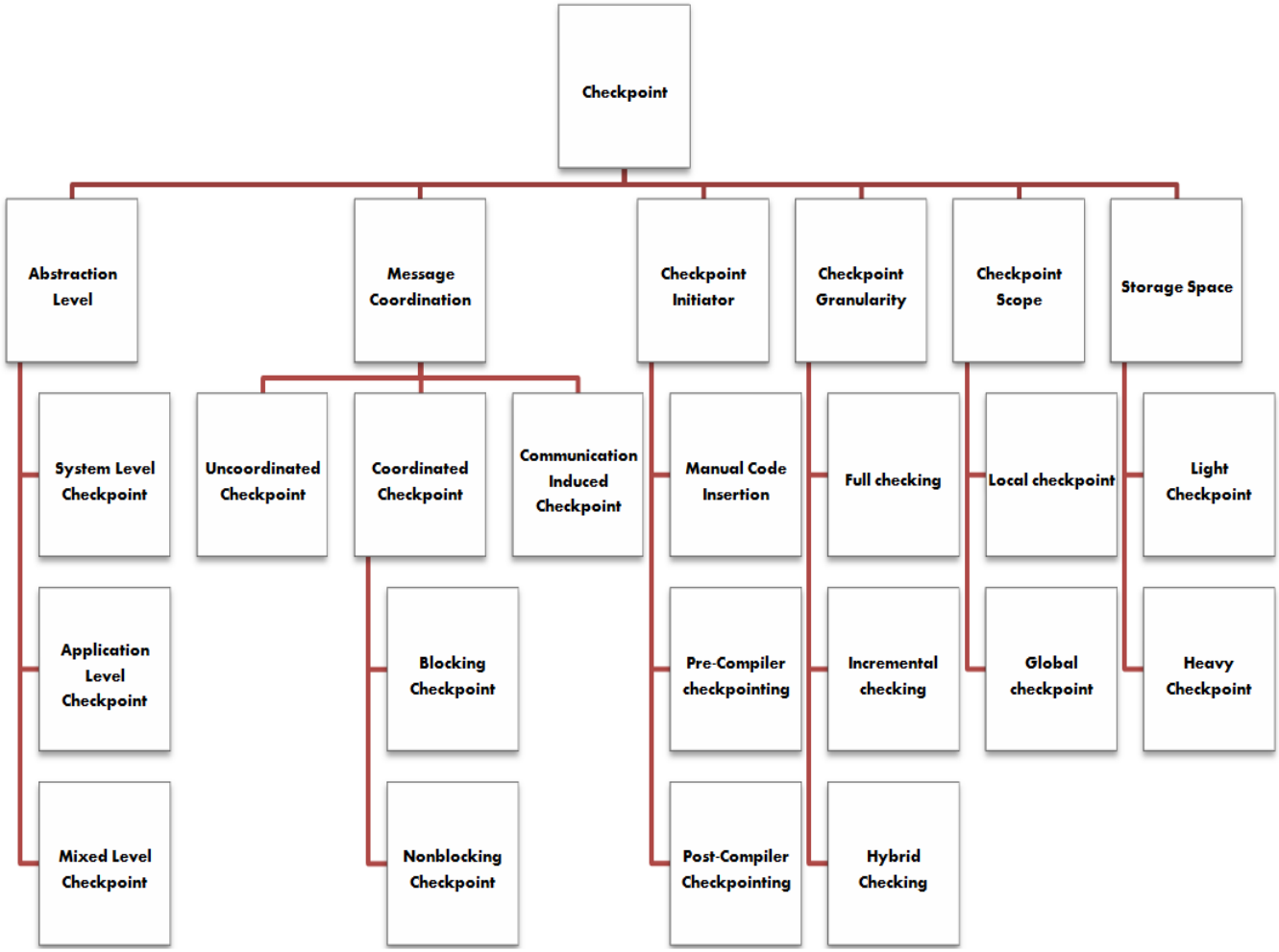

Figure 2 – Classification of the Checkpoint

A. Abstraction Level

Abstraction level at which the application current state is saved is the criteria for categorization. There are three kinds under this categorization

- **Checkpoint of System Level Checkpoint**: System level checkpoint method is automatic and transparent check-pointing of application at the operating system or middleware level is provided.
- **Checkpoint of Mixed Level**: Grouping of User Level Checkpoint and System Level Checkpoint is called as checkpoint of mixed level.
- **Application or User Level Checkpoint**: The Fault Tolerance is accomplished by application within itself by provided that self-containing code.

B. Message Coordination:

Processes of system handle an orphan message and in-transit is the criterion for this categorization. The following are the types of message coordination [19]

- **Coordinated / Synchronous Checkpoint:** This approach is to maintain process of the consistent global checkpoint and its follow two phase commit. In first phase, the tentative checkpoints are taken and made permanent in second phase. The processes are rolling back on fault to permanent checkpoint.
- **Uncoordinated/Asynchronous Checkpoint:** without coordinate with the other process, every process of the application takes the checkpoint separately.

i. **Check-pointing of Blocking**: The process leftovers blocked, until the whole check-pointing activity is finish after intriguing a local checkpoint to prevent an orphan message. To continue its execution as soon as it finishes its local checkpoint by this process is permitted. Limitation is computation is blocked throughout the check-point process.

ii. **Check-pointing of Non-blocking**: At the time of local checkpoint it may survive in transit and an orphan message. Whereas taking checkpoints, the processes not require stopping their execution. Limitation of this process, to prevent the process from receive an application message that would result is inconsistent checkpoint.

- **Checkpoint of Communication Induced:** Implement at the formation of global checkpoint i.e., uncoordinated by this method. By independently the local checkpoints have been formed. Still, by forcing additional checkpoints evaded the domino effect, so as to ensure the global checkpoint ultimate progress**.**

C. Checkpoint Initiator:

A process that initiates check pointing is called as simply initiator or initiator process. Initiator decides when to initiate check pointing procedure. Under this there are three kinds

- **Manual:** manually initiate a checkpoint by calling some user defined functions in the workflow code.
- **Pre-compiler checkpoint:** It works at the variable level are being user-directed, that the engendering small checkpoint files. It permits parallel process to checkpoint separately, without message-logging or runtime coordination.
- **Post-Compiler Checkpoint:** After patching all structure checks, such as checkpoint validation, we found some logical bug; once again process the checking process using post-compiler check pointing.

D. Checkpoint Granularity

The most extraordinary property of common check pointing techniques is granularity. Under this classification there are three kinds.

- **Full checkpoint:** A full checkpoint is a traditional checkpoint mechanism which intermittently saves the total state of the application to the local storage.
- **Incremental checkpoint:** In the Incremental checkpoint method, the first checkpoint is typically a full checkpoint. After that, only modified pages are check pointed at some predefined interval.
- **Hybrid Checkpoint:** This technique exchange between incremental and full checkpoints: On the incremental checkpoints, just data changed as the last checkpoint is captured.

E. Checkpoint Scope

- **Local checkpoint:** In the distributed systems, save the local states of all processes at convinced instants of time respectively. This saved state is called as local checkpoint. It is the saved state process at the processor at a certain instance.
- **Global checkpoint:** It is set of local checkpoints, that one from the each process.

F. Storage Space

Storage Checkpoint features that promptly generates a persistent image of the file system at an accurate point in time. Storage Checkpoints offers: Persistence through crashes and reboots, capability of data to be instantly writeable by preserve the directory hierarchy, the file system metadata, and the user data.

- **Lightweight checkpoint*:*** In lightweight check pointing the intermediary data is not saved; only a reference to it is saved presumptuous so the storage is reliable.
- **Heavyweight checkpoint*:*** heavyweight check pointing saves the intermediate data all along with their other things in the checkpoint.

VI. FAULT TOLERANCE CHALLENGES

Require suspicious analysis and consideration because of their inter-dependability, complexity and the subsequent reasons to provide the fault tolerance. [20][21]:

- To localize the faults, cloud heterogeneity is the main obstruction. Requires implementing the automatic fault tolerance technique for many instances of application are running on a number of VM.
- For creating a reliable system, various technologies from computing various vendors of cloud infrastructure require to be integrated.
- It is difficult to design an optimal fault tolerance solution because of limited information is provided to the users due to the high system complexity.
- Automatic fault tolerance has to respond for synchronization among different clouds.
- There is more likelihood of errors because processing is complete on remote computers.
- To ensure high availability and reliability of multiple clouds computing providers with sovereign software stacks must be used.
- For assess fault tolerance performance compone nt in evaluation with similar ones, a benchmark based method can be developed in cloud environment.
- Cloud environment are scalable with unpredicted, frequently virtualized resources and dynamism are give as a service, so it is complex to understand the state of changing system.

VII. CONCLUSION

Cloud computing environment is dynamic which ensuing in failures and faults because of unpredicted system behavior. Reliability and availability of the systems are of the most important requirements. FT system is one of the foremost parts of any system as it guarantees the system to continue its working during fault or failure. To enhance the system, there are so many fault tolerance models to present different fault tolerance mechanisms. However, which require some concern for every model or frame work because there are many challenges. No one of them can full fill the all aspects of faults still there are some drawback. Therefore, possibility to overcome the limitation of all previous model, make an effort of compact model which will envelop the maximum fault tolerance aspect. This paper discussed the concept of the fault tolerance, the fault tolerance techniques, illustrates various states of checkpoint based on fault tolerance techniques the art fault tolerant strategies for the cloud environments.

VIII. REFERENCES

[1]. C. S. Buyya, R. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Vol. 25, No. 6, pp. 599-616, June 2009.

[2]. S. Prathiba and S. Sowvarnica, "Survey of failures and fault tolerance in cloud," in *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, 2017, pp. 169–172.

[3]. Mladen A. Vouk, "Cloud Computing – Issues, Research and Implementations", Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA,Journal of Computing and Information Technology - CIT 16, 2008, 4, 235–246 doi:10.2498 /cit.100139

[4]. V.M.Sivagami, K.S.EaswaraKumar, "Survey on Fault Tolerance Techniques in Cloud Computing Environment", International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-9, December 2015.

[5]. Ms. Disha H. Parekh, Dr. R. Sridaran, "An Analysis of Security Challenges in Cloud Computing," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.1, 2013.

[6]. Zhengxiong Hou, Xingshe Zhou, Jianhua Gu,Yunlan Wang,Tianhai Zhao, "ASAAS: Application Software as a Service for High Performance Cloud Computing," 12th IEEE International Conference on High Performance Computing and Communications, 2010.

[7]. Wei-Tek Tsai, Xin Sun, Janaka Balasooriya, "Service-Oriented Cloud Computing Architecture," Seventh International Conference on Information Technology, 2010.

[8]. Abhishek Bhavsar, Ameya More, "A Holistic Approach to Autonomic Self-Healing Distributed Computing System" International Journal of Computer Applications (0975 – 8887) Volume 76–No.3, August 2013.

[9]. Harpreet Kaur, Amritpal Kaur, "A Survey on Fault Tolerance Techniques in Cloud Computing", International Journal of Science, Engineering and Technology, Volume 3 Issue 2, 2015.

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

[11]. http://www.cs.ucla.edu/~rennels/article98.pd f

[12]. A. M. Saleh and J. H. Patel, "Transient-Fault Analysis for Retry Techniques," *IEEE Trans. Reliab.*, vol. 37, no. 3, pp. 323–330, 1988.

[13]. Elvin Sindrilaru,,Alexandru Costan,, Valentin Cristea," Fault Tolerance and Recovery in Grid Workflow Management Systems", 2010 International Conference on Complex, Intelligent and Software Intensive Systems.

[14]. Anju Bala, Inderveer Chana, "Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing", IJCSI international Journal of Computer Science Issues, Vol.9, Issue 1, No 1, January 2012

[15]. Benjamin Lussier, Alexandre Lampe, Raja Chatila, Jérémie Guiochet, Félix Ingrand, Marc-Olivier Killijian, David Powell, "Fault Tolerance in Autonomous Systems: How and How Much?" LAAS-CNRS 7 Avenue du Colonel Roche, F-31077 Toulouse Cedex 04, France

[16]. Jean-clande Laprie "Dependable computing and fault tolerance: concepts and terminology" LAAS-CNRS 7 Avenue du Colonel Roche, 31400 Toulouse, France

[17]. Patel et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(12), December - 2013, pp. 573-576

[18]. Wenbing Zhao, P.M. Melliar and L.E. Mose, " Fault Tolerance Middleware for Cloud Computing", 2010 IEEE 3rd International Conference on Cloud Computing.

[19]. Raman Kumar, Dr. Parveen Kumar ,"Review of Some Checkpointing Schemes for Distributed and Mobile Computing", Int. J. Advanced Networking and Applications Vol: 06 Issue: 06 (2015) ISSN: 0975- 0290.

[20]. AnjuBala, InderveerChana, "Fault Tolerance-*Challenges, Techniques and Implementation in Cloud Computing"*, IJCSI International Journal of Computer Science Issues,January 2012.

[21]. Pawan Thakur RoohiAli,"Cloud Computing Architecture" in *Cloud Computing 1st Edition New Delhi India Tech India Publication Series 2013-14*