Research Directions towards Fault Tolerance in Cloud

Moin Hasan

School of Computer Science and Engineering,

Lovely Professional University, Phagwara, India.

Abstract: Cloud computing has been the most prominent Internet-based computing service paradigm for more than a decade now. It has gained enormous popularity in various sectors like business, academics, and research, etc. Cloud user base has been increasing day by day which requires the robust and efficient fault tolerance strategies to ensure the seamless and reliable service provisioning. Therefore, among various research challenges in the cloud environment, fault tolerance is considered as one of the most focused. Implementation of fault tolerance in cloud is a challenging task due to its ever-growing complexity and architectural dynamicity. Numerous research works have been proposed and implemented in this area, but still, there are certain challenges associated which need further exploration. This paper is an endeavour to enlist and discuss those challenges in detail.

1. Introduction

Cloud computing has gathered considerable attention towards the provisioning of IT resources via the Internet. According to [1], cloud computing is a paradigm which facilitates the on-demand access to a shared group configurable resources in a convenient manner. It is majorly characterized by its capabilities of on-demand self-service, global access, resource pooling, and swift elasticity, etc. Service/resource provisioning in cloud can be broadly graded as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) subject to the type of resources included [2]. However, as the domain of cloud resources has been increased noticeably, cloud is also considered as an Everything as a Service (XaaS) model. The above cloud model can be deployed in the form of public cloud, private cloud, and community cloud. Moreover, multiple deployment models can be combined to form a hybrid cloud. Figure 1 shows a basic cloud architecture.
Since its evolution, cloud is one of the most contributed research areas in the field of computer science and engineering. Various research challenges in cloud include fault tolerance [3], load balancing [4], energy management [5], resource reputation [6], etc. Due to the Internet-based resource provisioning, ever-growing complexity, and architectural dynamicity, cloud environment is prone to various types of faults and often termed as an adolescent computing paradigm yet now in terms of its fault tolerance capabilities[7]. Therefore, among all, fault tolerance is considered as one of the most focused research areas in cloud.

Plentiful research has been carried out towards fault tolerance in cloud for more than a decade now. However, there are numerous challenges which are wide open and can be considered for further research contribution. Therefore, this paper is an endeavour to enlist and discuss those challenges in detail in order to felicitate the young researchers working (or willing to work) towards fault tolerance in cloud.

The paper is organized as follows: Section 2 describes various fault types in cloud. Section 3 describes various fault tolerance techniques adopted in cloud. Section 4 includes and discusses various research challenges towards fault tolerance in cloud. The paper is finally concluded in Section 5.
2. Types of Faults in Cloud

Since, cloud is a computing paradigm which follows the distributed architecture, therefore, it inherits the faults which occur in the conventional distributed computing architecture [8]. Broadly, faults which occur in cloud are classified into two groups viz., crash faults and byzantine faults. Crash faults occur as the resultant of the failure of system component(s) (like processing units, storage disks, power supplies, etc.). Byzantine faults are basically characterized as the unexpected output either due to any ambiguity in the software program or due to unpredictable conditions. The faults which are considered under the category of crash faults are described as follows:

i. Hardware fault: It occurs due to the breakdown of hardware module(s) in the system and disturbs the infrastructure layer of the cloud service delivery model.

ii. Network fault: It occurs due the breakdown of network component(s) in the cloud (like routers, switches, access points, etc.). Similar to hardware fault, it also disturbs the infrastructure layer.

iii. Configuration fault: It is the resultant of disturbed ordering of system components. It also affects the infrastructure layer.

The faults which are considered as byzantine faults are described as follows:

i. Software fault: It generally occurs when there is some error or exception in the software program. It may also occur when the software update takes place.

ii. Parametric fault: It is the resultant of unspecified deviation(s) in the system parameters.

iii. System fault: It the generally the resultant of insufficient awareness of the control procedures used for the service orchestration in cloud.

iv. Time-constraint fault: When a running application missed its specified deadline of completion, the situation is called as time-constraint fault.

Apart from the above discussed faults, there are certain faults which are difficult to be categorized as either crash faults or byzantine faults. They are described as follows:

i. Constraint fault: It occurs due to irresponsible behaviour of the agent where a faulty condition arises but ignored.

ii. Participant fault: It is the resultant of the conflict between similar or different cloud participants (like cloud user, cloud provider, administrator, etc.).
iii. Resource-contention fault: When a conflict occurs due to the sharing of resources in cloud, the resultant condition is known as resource-contention fault.

iv. Retrospective fault: It generally arises due to insufficient or inappropriate knowledge about the earlier behaviour of the system.

v. Stochastic fault: It is the resultant when the system lacks statistical information to assess the system state.

The above faults can be considered in either of the two categories (crash faults or byzantine faults). Since, cloud follows a layered architecture, therefore, it can be concluded that faults which erupt in a specific layer (IaaS/PaaS/SaaS) may disturb the layer(s) above as well. Meaning thereby, fault occurrence in PaaS may disturb the SaaS and consequently may interrupt the service delivery. For example, faulty operating system (at the PaaS layer) may interrupt the installed software (at the SaaS layer). See Figure 2 for further description.

![Figure 2. Fault Relationship in Cloud](image)

3. Techniques for Fault Tolerance in Cloud

There are numerous techniques proposed and implemented to provide fault tolerance in cloud as well other distributed computing models. However, these techniques are broadly classified as proactive techniques and reactive techniques. Proactive techniques are those where fault occurrence is predicted using various Artificial Intelligence algorithms and methods [9]. After the fault prediction, essential measures are taken to abridge the fault occurrence and keep the system uninterrupted. Eminent proactive fault tolerance techniques adopted in cloud are pre-emptive migration, self-healing, and system rejuvenation [10] discussed in Section 3.1. On the other hand, reactive techniques apply specific fault tolerance procedures after the occurrence of fault [9].
Popular reactive fault tolerance techniques used in cloud are job migration, checkpoint restart, and replication [10] discussed in Section 3.2.

3.1. Techniques for Proactive Fault Tolerance
Following are the popular proactive fault tolerance techniques adopted in cloud computing environment:

3.1.1. Pre-emptive Migration
In this technique, AI-enabled system monitoring procedures are applied (continuously or periodically) to predict the faults. Whenever a fault is predicted, the application running on the fault-predicted resource is migrated to another safer resource.[14]–[16].

3.1.2. Self-Healing
This technique is based on the biological fact, “How the living organisms manage to survive during difficult situations [11], [12].” In this technique, system is made capable to get recovered autonomously from the faults. Deployment of self-healing requires implementation of supervision tasks to develop specific fault recovery procedures [13]. After the deployment, system becomes capable of perceiving the faulty conditions and to apply the specific recovery procedures.

3.1.3. System Rejuvenation
In this technique, system backup is taken periodically. After taking the backup, system state is cleaned and repaired in order to obtain a fresh (rejuvenated) system state. Depending upon the system, two types of system rejuvenation are full system rejuvenation and partial system rejuvenation. Depending upon the time period, two modes of system rejuvenation are fixed time rejuvenation and variable time rejuvenation.

3.2. Techniques for Reactive Fault Tolerance
Popular reactive fault tolerance techniques used in cloud are discussed as follows:

3.2.1. Job Migration
In this technique, the application running on a faulty resource is migrated to some other healthy resource in the system [18]. It is similar to pre-emptive migration technique except the fact that migration of the application takes place after fault occurrence.
3.2.2. **Checkpoint Restart**

It involves the process of saving the running state of applications periodically in a safe checkpoint repository. Whenever a fault occurrence interrupts any running application, the application can be restarted from its last saved state [17].

3.2.3. **Replication**

It is the most common fault tolerance technique used in any distributed computing paradigm. In this technique, multiple resources are allocated to form a replication group for a single application [19]. In case, one resource in the replication group becomes faulty, the running application will be continued on the remaining resources.

Figure 3 shows a hierarchical distribution of various fault tolerance techniques adopted in cloud.

![Figure 3. Hierarchy of Fault Tolerance Techniques](image)

4. **Research Challenges towards Fault Tolerance in Cloud**

This Section describes various challenges generally encountered towards fault tolerance in cloud. This Section also gives the possible directions to carry out the research in order to overcome the specified challenges.

i. **Improvement of task completion rate:** Besides witnessing enormous research towards fault tolerance in cloud, to the best of the knowledge, no research work has claimed 100% task completion rate. In the research works [20], [21], authors claimed that they have achieved about 95% task completion rate. It means that about 5 of 100 users remain unsatisfied. In case of commercial public clouds where millions of users access cloud services, this number would turn out to be several thousands. Therefore, improvement of task completion rate through robust fault tolerance mechanism can be considered as a good direction of research.
ii. Optimization of overhead in proactive fault tolerance techniques: Proactive fault tolerance techniques require critical examination and monitoring of the system state which in turn contributes towards overall fault tolerance overhead. Research can be carried out in this area so that efficient system monitoring mechanisms can be evolved.

iii. Provision of hybrid fault tolerance: Available fault tolerance techniques are either proactive or reactive. However, it would be highly appreciable if the advantages of both proactive and reactive fault tolerance techniques can be combined to have a hybrid fault tolerance technique.

iv. Application of pre-emptive migration technique as an auxiliary fault tolerance technique: Checkpoint restart is often applied as a supplementary fault tolerance technique along with replication. However, application of pre-emptive migration as an auxiliary technique would improve the fault tolerance comparatively higher because pre-emptive migration ensures zero loss in the computation.

v. Implementation of fault tolerance techniques capable of tolerating each type of fault: Generally, the available fault tolerance mechanisms are capable of tolerating only a limited number of faults. To the best of the knowledge, no author has yet claimed to have a fault tolerance covering complete fault applicability.

vi. Fault tolerance provisioning through load balancing: Many of the faults occur due to load imbalance in the system. Load imbalance may occur due to excessive resource consolidation in the greed of cost saving. To have an efficient load balancing mechanism would considerably decrease the probability of fault tolerance.

vii. Provisioning of flexible fault tolerance in cloud: Cloud is on-demand service provision model. However, the available fault tolerance mechanisms look reluctant to consider fault tolerance associated with the users’ demand. Flexible fault tolerance where the users can opt the level of fault tolerance as per their necessity will further emphasize the service-oriented model of cloud.

Apart from the above challenges, literature shows that researchers are more focused about fault tolerance rather than fault detection. However, an efficient fault detection mechanism will ultimately improve the fault tolerance of the system.

5. Conclusions
In the current scenario, cloud serves millions of users across the globe. Among various issues, fault tolerance is considered as one the most important to have a reliable service orchestration. Besides having extreme research contribution, various challenges are still wide open inviting further research efforts. This paper highlighted those challenges provided several research directions towards fault tolerance in cloud. Along with the research directions, a conceptual study of fault types and fault tolerance techniques in cloud is also given in the paper.

References


