A Survey of Cloud Computing Fault Tolerance: Techniques and Implementation

ABSTRACT

Cloud computing provides services as a type of Internet-based computing using data centers that contain servers, storage and networks. For this reason, the cloud computing its great potentials in low cost and on-demand services. In recent years, the end user is highly increased to utilize the services in cloud computing. However, the faulty of infrastructure, software and application are the major problem in cloud computing. Fault tolerance uses techniques that concerned to guarantee availability, reliability of critical services and application execution. This paper discusses the existing fault tolerance techniques and challenges to minimize failure impact on the system and application execution in cloud computing.

Keywords

Cloud Computing, Fault Tolerance, Data Centers.

1. INTRODUCTION

Cloud computing can be defined as provide a service over internet by using computational resources such as storages, operating systems etc[1]. The elementary advantage of cloud computing is emerging as a new computing paradigm which aims to provide reliable, low costs, high availability, scalability and elasticity for end-users [2]. Amazon is the first company to look into the growing importance of Cloud computing very seriously followed by Google and IBM [3]. Many companies such as Google, Microsoft and Salesforce are delivering services from the cloud . Google Has a private cloud that it uses for delivering many different services to its users, including statistics, analytics, text translations, and much more services based on big data analytics [18, 19]. Microsoft has online service that allows for content and business intelligence tools to be moved into the cloud, and Microsoft currently makes its office applications available in a cloud. Salesforce Runs its application set for its customers in a cloud, and it's Force.com and Vmforce.com products provide developers with platforms to build customized cloud services.

Once a cloud is established, the cloud computing services are deployed in terms of business models can differ depending on requirements. The primary service models being deployed are commonly known as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). SaaS defined as consumers purchase the ability to access and use an application or service that is hosted in the cloud. PaaS defined as consumers purchase access to the platforms, enabling them to deploy their own software and applications in the cloud. IaaS means the consumers control and manage the systems in terms of the operating systems, applications, storage, and network connectivity, but do not themselves control the cloud infrastructure.

The reliability of Cloud computing still is the major problem in cloud computing and concerned by among users. Due to economic pressures, these computing infrastructures often use commodity components exposing the hardware to scale and conditions for which it was not originally designed [4]. Fault tolerance computing means the job can continue to correctly perform all tasks when hardware and/or software failures. Indeed, applications require fault tolerance abilities so that they can overcome the impact of system failures and perform their functions correctly when failures happen. The main benefits of implementing fault tolerance in cloud computing include failure recovery, improving reliability, and enhanced availability [5]. The motivation of the survey of existing fault tolerance techniques and models in cloud computing is to encourage researcher to contribute in developing more efficient algorithm. This paper is planned to deliberates about various aspect of faults and the need of fault tolerance in cloud computing. This paper aims deliberates about various aspect of faults and the need of fault tolerance in cloud computing.

2. CLOUD COMPUTING

The NIST definition is one of the clearest and most comprehensive definitions of cloud computing and is widely referenced in US government documents and projects. This definition describes cloud computing as having five essential characteristics, three service models, and four deployment models. The essential characteristics are [6]:

- On-demand self-service: computing resources can be acquired and used at any time without the need for human interaction with cloud service providers. Computing resources include processing power, storage, virtual machines etc.
- Broad network access: the previously mentioned resources can be accessed over a network using heterogeneous devices such as laptops or mobiles phones.
- Resource pooling: cloud service providers pool their resources that are then shared by multiple users. This is referred to as multi-tenancy where for example a physical server may host several virtual machines belonging to different users.
- Rapid elasticity: a user can quickly acquire more resources from the cloud by scaling out. They can scale back in by releasing those resources once they are no longer required.
- Measured service: resource usage is metered using appropriate metrics such monitoring storage usage, CPU hours, bandwidth usage etc.

Figure 1 provides an overview of the common deployment and service models in cloud computing, where the three service models could be deployed on top of any of the four deployment models. Vaquero et al. studied 22 definitions of cloud computing and proposed the following definition: Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing also for optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs. This definition includes three of the five characteristics of cloud computing described by NIST, namely resource pooling, rapid elasticity and measured service but fails to mention on-demand self-service and broad network access.

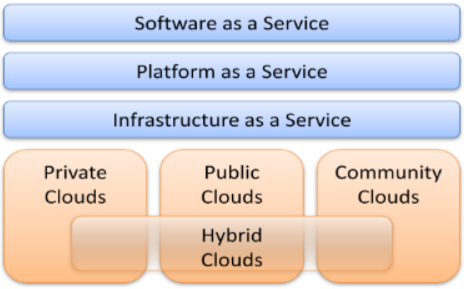

Fig1: Cloud computing deployment and service models [7].

Figure 1 illustrates different types of cloud Private, Public, Community and Hybrid. Private Clouds are typically owned by the respective enterprise and / or leased. Functionalities are not directly exposed to the customer, though in some cases services with cloud enhanced features may be offered – this is similar to (Cloud) Software as a Service from the customer point of view. Example: eBay. Public Clouds enterprises may use cloud functionality from others, respectively offer their own services to users outside of the company. Providing the user with the actual capability to exploit the cloud features for his / her own purposes also allows other enterprises to outsource their services to such cloud providers, thus reducing costs and effort to build up their own infrastructure. As noted in the context of cloud types, the scope of functionalities thereby may differ. Example: Amazon, Google Apps, Windows Azure. Hybrid Cloud, though public clouds allow enterprises to outsource parts of their infrastructure to cloud providers, they at the same time would lose control over the resources and the distribution management of code and data. In some cases, this is not desired by the respective enterprise. Hybrid clouds consist of a mixed employment of private and public cloud infrastructures so as to achieve a maximum of cost reduction through outsourcing whilst maintaining the desired degree of control over e.g. sensitive data by employing local private clouds.

3. TECHNIQUES AND IMPLEMENTATION OF FAULT TOLERANCE

There are various faults which can occur in cloud computing. These faults can be classified on several factors such as [8,9]:

- Network fault: A Fault occur in a network due to network partition, Packet Loss, Packet corruption, destination failure, link failure, etc.
- Physical faults: This Fault can occur in hardware like fault in CPUs, Fault in memory, Fault in storage, etc.
- Media faults: Fault occurs due to media head crashes.
- Processor faults: fault occurs in processor due to operating system crashes, etc.
- Process faults: A fault which occurs due to shortage of resource, software bugs, etc.
- Service expiry fault: The service time of a resource may expire while application is using it.

Also, a fault can be categorized on the basis of computing resources and time. A failure occurs during computation on system resources can be classified as: omission failure, timing failure, response failure, and crash failure. The Fault may be [10]:

- Permanent: These failures occur by accidentally cutting a wire, power breakdowns and so on. It is easy to reproduce these failures. These failures can cause major disruptions and some part of the system may not be functioning as desired.
- Intermittent: These are the failures appears occasionally. Mostly these failures are ignored while testing the system and only appear when the system goes into operation. Therefore, it is hard to predict the extent of damage these failures can bring to the system.
- Transient: These failures are caused by some inherent fault in the system. However, these failures are corrected by retrying roll back the system to previous state such as restarting software or resending a message. These failures are very common in computer systems.

4. FAULT TOLERANCE TECHNIQUES

There are various techniques available to provide fault tolerance as shown in Table 1. Below given techniques that achieve fault tolerance:

- Check pointing–It is an efficient task level fault tolerance technique for long running and big applications .In this scenario after doing every change in system a check pointing is done. When a task fails, rather than from the beginning it is allowed to be restarted that job from the recently checked pointed state.
- Job Migration – Sometimes it happens that due to some reason a particular machine fails and cannot execute job. On such a failure, a task is migrated to working machine using HA-Proxy. Also, there are algorithms that automatically determine the fault and migrates batch applications within a cloud of multiple datacenters.
- Self- Healing- In this method divide and conquer technique is used, in which a huge task is distributed into several parts. This division is done for better performance. In this, various instances of an application are running on various virtual machines and failure of all these individual instances are handled automatically.
- Safety-bag checks: In this case the blocking of commands is done which are not meeting the safety properties.
- S-Guard- It is less turbulent to normal stream processing. S-Guard is based on rollback recovery. SGuard can be implemented in HADOOP, AmazonEC2.

- Retry- In this case we implement a task again and gain. It is the simplest technique that retries the failed task on the same resource.
- Task Resubmission- A job may fail now whenever a failed task is detected, In this case at runtime the task is resubmitted either to the same or to a different resource for execution.
- Replication- Replication means copying. Several replicas of tasks are created and they are run on different resources, for effective execution and for getting the desired result. Hadoop, HA-Proxy, Amazon EC2 like tools are there on which replication can be implemented.

Table 1: Summary of Fault Tolerance Techniques

Fault Tolerance Techniques	*Polices*	*System*	*Programming Framework*	*Environment*	*Fault Detected*	*Application Type*
Self Healing, Job Migration, Replication	Reactive/ Proactive	HAProxy[13]	Java	Virtual Machine	Process/node failure	Load balancing Fault Tolerance
Check pointing	Reactive	SHelp[12]	SQL, JAVA	Virtual Machine	Application Failure	Fault Tolerance
Check pointing, Retry, Self Healing	Reactive/ Proactive	Assure[9]	JAVA	Virtual Machine	Host, Network failure	Fault Tolerance
Job Migration, Replication, Sguard, Resc	Reactive/ Proactive	Hadoop[7]	Java, HTML, CSS	Cloud Environment	Application/ node failure	Data intensive
Replication, Sguard, Task Resubmission	Reactive/ Proactive	AmazonEC2[8]	Amazon Machine Image, Amazon Map	Cloud Environment	Application/ node failure	Load balancing, fault Tolerance

Also, there are mainly three different types of replication schemes such as Active Replication, Semi-Active Replication and Passive Replication

- Masking- After occupation of error recovery the new state needs to be identified as a transformed state. If this process applied systematically even in the absence of effective error provide the user error masking [11].
- Resource Co-allocation- In refers to the process of allocating resources for further execution of task. Many algorithms are designed, that deals which resource allocation depending on the properties of VM such as workload, type of task, capacity of VM, energy awareness etc.
- Rescue Workflow- A workflow consists of a sequence of connected steps where each step follows without delay or gap and ends just before the subsequent step may begin. In this technique, it allows the workflow to carry on until it becomes unimaginable to move forward without catering the failed task.
- User Specific (defined) Exception Handling- In this case, whenever fault is detected, action is predefined by the user, i.e. user defines the particular treatment for a task on its failure.

5. FAULT TOLERANCE MODELS

- Table 2 illustrates several models that are implemented based on above techniques are as follows:
- AFTRC – It is an Adaptive Fault Tolerance model in Real time Cloud Computing. In this proposed model system tolerates fault proactively and makes decision on the basis of the reliability of the processing nodes [12].
- LLFT - is a propose model which contains a low latency fault tolerance (LLFT) middleware for providing fault tolerance for distributed applications deployed with in the cloud computing environment. This middleware replicates application by the using of semi-active replication or semi-passive replication process to protect the application against various types of faults [13].
- FTM- is a model to overcome the limitation of existing methodologies and achieves the reliability and flexibility, they propose an inventive perspective on creating and managing fault tolerance .By this particular methodology user can specify and apply the desire level of fault tolerance. FTM architecture this can primarily be viewed as an assemblage of several web services components, each with a specific functionality [14].
- FTWS- is a Fault Tolerant Work flow Scheduling algorithm for providing fault tolerance by using replication and resubmission of tasked based on based on the priority of the task. This model is based on the fact that work flow is a set of tasks processed in some order based on data and control dependency. Scheduling the workflow along with the task failure consideration in a cloud environment is very challenging. FTWS schedule and replicates the tasks to meet the deadline.

Table 2: Summary of Fault Tolerance Models [17]

Model Name	**Regular Protection against type of fault**	**Application procedure for tolerate the fault**
AFRTC	Reliability	1. Delete node depending on their reliability 2. Back word recovery with the help of check pointing
LLFT	Crash-cost, trimming fault	Replication
FTM	Reliability, availability, on demand service	Replication users application and in the case of replica failure use algorithm like gossip based protocol.
FTWS	Dead line of work flow	Replication and resubmission of jobs
Candy	Availability	1. It assembles the model components generated from IBD and STM according to allocation notation. 2. Then activity SNR is synchronized to system SNR by identifying the relationship between action in activity SNR and state transition in system SNR
FT-Cloud	Reliability, crash and value fault	1. Significant component determined based on the ranking. 2. Optimal ft technique is determined.

- Candy- is a component based availability model. It is based on the high availability assurance of cloud service is one of the main characteristic of cloud service and also one of the main critical and challenging issues for cloud service provider [15].
- FT-Cloud- is a component ranking based frame work and its architecture for building cloud application. FTCloud occupies the component invocation structure and frequency for identify the component. Also, there is an algorithm to automatically govern fault tolerance stately [16].

6. CONCLUSION

Fault tolerance is one of the main challenges and critical issue in cloud computing. It is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. This paper identified some of fault tolerance algorithms which distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Presently, there are several mechanisms for fault tolerance but still there are number of challenges which need to be considered. Also, there are some drawbacks no one of them can fulfill all the aspects of faults. So, there is likelihood to overcome these drawbacks and try to make a solid model that may cover maximum fault tolerance aspect.