ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 8 August, 2014 Page No. 7938-7942

Fault Detection and Tolerant System (FDTS) for SaaS Layer in Cloud Computing

Shweta Jain[1], Prof. Ashok Verma[2], Prof. Rashween Kaur Saluja[3]

[1]Dept. of CSE, GGITS, Jabalpur,India
jn_shweta06@yahoo.co.in

[2]Head of the Dept. of CSE, GGITS, Jabalpur,India
vermaashok33@gmail.com

[3]Asst. Prof. Dept. of CSE, GGITS, Jabalpur,India
rashweensaluja@ggits.org

Abstract: *The increasing popularity of Cloud computing as an attractive alternative to classic information processing systems has increased the importance of its correct and continuous operation even in the presence of faulty components. Fault tolerance is a major concern to guarantee availability and reliability of critical services as well as application execution. In order to minimize failure impact on the system and application execution, failures should be anticipated and proactively handled. Fault tolerance techniques are used to predict these failure and take an appropriate action before failure actually occur. In this paper, we introduce an innovative, system-level, modular perspective on creating and managing fault tolerance in Clouds. We propose a high-level approach at SaaS layer for hiding the implementation details of the fault tolerance techniques to application developers and users. In particular, the service layer hides the user from fault tolerance mechanism, and does not require knowledge about the fault tolerance technique applied and that are available in the Cloud and their implementations.*

The fault tolerant technique applied shall use the heartbeat algorithm(s) and gossip algorithm to detect whether the application is working smoothly or not. In case the application is detected to be down then the proposed work deploys an application recovery mechanism applied at the SaaS layer, which will try to start, recover from failure or will reinstall the application so that users can use the same smoothly with minimum downtime.

Keywords: heartbeat algorithm, gossip algorithm, time synchronization, fault tolerance, SaaS Layer development.

1. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers.

Fault tolerance bear–on with all the inevitably techniques to enable robustness and dependability. The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics.

Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of cloud ccomputing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. Recent downtime of Amazon's S3 is such an example.

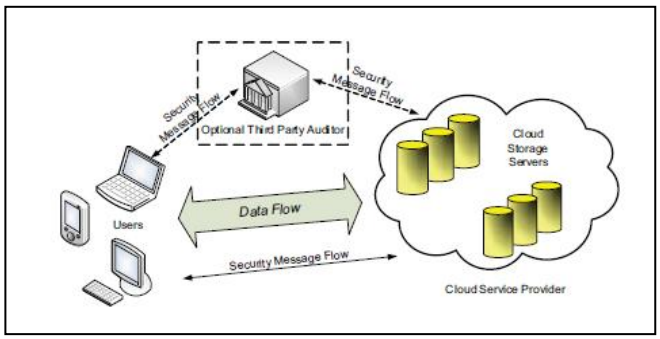


Figure 1: Cloud Data Storage Architecture

1.2. Computing as a Service

One of the main tenets of Cloud Computing is the 'as-a-Service' paradigm in which 'some' service is offered by a Service Provider (also known as a Cloud Service Provider) to a User (consumer) for use. This service can also be categorized according to the application domain of its deployment. Examples of application domains that offer services are: Financial e.g. Mint.com, Managerial e.g. Ever Note and Analytical e.g. Google Analytics. The agreed terms of use, indicating the actions that must be taken by both the provider and consumer, are described in a contract that is agreed upon before service provision. Failure to honor this agreement can lead to denial of service for the consumer or legal liability for the service provider. This contract is often described as a Terms of Service or Service Level Agreement. Moreover, as part of this agreement the service provider will provide a Privacy Policy which outlines how the user's data will be stored, managed, used and protected.

1.2.3. Service Levels

The services offered are often categorized using the SPI Service Model. This model represents the different layers/levels of service that can be offered to users by service providers over the different application domains and types of cloud available. Clouds can be used to provide as-a-Service: software to use, a platform to develop on, or an infrastructure to utilize. Figure 2 summarizes the SPI Service Model.

Software as a Service: The first and highest layer is known as Software as a Service (SaaS). It represents the applications that are deployed or enabled over a cloud by CSPs.

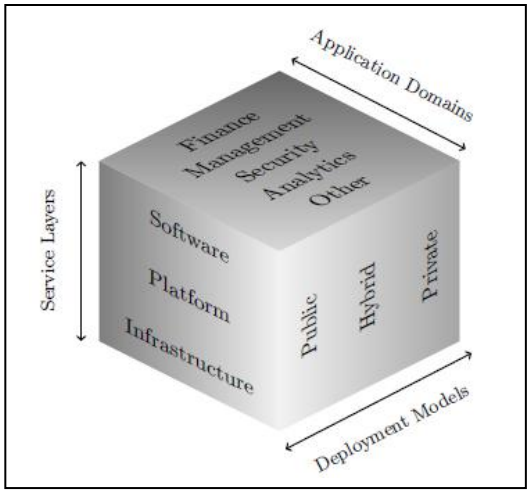


Figure 2: Summary of the SPI Service Model

Applications that often offer an API to allow for greater application extensibility. For instance, Google Docs can be seen as the archetypal SaaS application, it has been deployed solely within the Cloud and offers several APIs to promote use of the application.

Platform as a Service: The next layer is known as Platform as a Service (PaaS). This represents a development platform that developers can utilize to write, deploy and manage applications that run on the cloud. This can include aspects such as development, administration and management tools, run-time and data management engines, and security and user management services. For instance, Force.com and Amazon Web Services offers a suite of services that allows developers to construct an application that is deployed using web-based tooling.

Infrastructure as a Service: The final and lowest layer is known as Infrastructure as a Service (IaaS). CSP offer developers, a highly scaled and elastic computing infrastructure that is used to run applications. This infrastructure can be comprised of virtualized servers, storage, databases and other items. Two well known examples are the Amazon Elastic Compute Cloud, a commercial platform offered as part of Amazon.com Web Service platform and Eucalyptus, an open source platform that offers the same functionality.

(a) Application Area

Cloud computing is a collection of technologies that allow IT resources to be virtualized, used on an on-demand basis and delivered via the Internet as services. Cloud computing can be considered a new computing paradigm in so far as it allows the utilization of a computing infrastructure at one or more levels of abstraction, as an on-demand service made available over the Internet or other computer network. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time and the service is fully managed by the provider. Because of the its features of greater flexibility and availability at lower cost, cloud computing is a subject that has been receiving a good deal of attention. Cloud Computing can be classified into 4 types on the basis of location where the cloud is hosted:-

Public Cloud: A public cloud is one in which the infrastructure and other computational resources that it comprises are made available to the general public over the Internet. It is owned by a cloud provider selling cloud services and, by definition, is external to an organization.

Private Cloud: A private cloud a proprietary network or a data center that supplies hosted services to a limited number of people. It may be managed either by the organization or a third party, and may be hosted within the organization's data center or outside of it.

Community Cloud: A community cloud is somewhat similar to a private cloud, but the infrastructure and computational resources are shared by several organizations that have common privacy, security, and regulatory considerations, rather than for the exclusive use of a single organization.

Hybrid Cloud: A hybrid cloud is a composition of two or more clouds (private, community, or public) that remain unique

entities but are bound together by standardized or proprietary technology that enables interoperability.

Cloud computing utilizes three delivery models by which different types of services are delivered to the end user. The three delivery models are the SaaS, PaaS and IaaS which provide infrastructure resources, application platform and software as services to the consumer.

1) Software-as-a-Service: SAAS is defined as a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network. Also known as "on demand" software, it is the most mature type of Cloud Computing because of its high flexibility, proven support services, enhanced scalability, reduced customer maintenance, and reduced cost due to their multi-tenet architectures. It is a model of software deployment whereby one or more applications and the computational resources to run them are provided for use on demand. Its main purpose is to reduce the total cost of hardware and software development, maintenance, and operations. Security provisions are carried out mainly by the cloud provider. The cloud subscriber does not manage or control the underlying cloud infrastructure or individual applications, except for preference selections and limited administrative application settings.

2) Platform-as-a-Service: PAAS provides infrastructure on which software developers can build new applications or extend existing applications without requiring the need to (purchase development, QA, or production server infrastructure. It is a model of software deployment where the computing platform is provided as an on-demand service upon which applications can be developed and deployed. Its main purpose is to reduce the cost and complexity of buying, housing, and managing the underlying hardware and software components of the platform, including any needed program and database development tools. The cloud subscriber has control over applications and application environment settings of the platform. Security provisions are split between the cloud provider and the cloud subscriber.

3) Infrastructure-as-a-Service: Infrastructure-as-a-Service (IaaS) is a model of software deployment whereby the basic computing infrastructure of servers, software, and network equipment is provided as an on-demand service upon which a platform to develop and execute applications can be established. Its main purpose is to avoid purchasing, housing, and managing the basic hardware and software infrastructure components, and instead obtain those resources as virtualized objects controllable via a service interface. The cloud subscriber generally has broad freedom to choose the operating system and development environment to be hosted Security provisions beyond the basic infrastructure are carried out mainly by the cloud subscriber.

2. PROBLEM STATEMENT

Because of fault tolerance difficulties, inter-dependability and the following reasons it requires careful consideration and analysis

- Different methodologies from different vendors of cloud environment need to be integrated for establishing a reliable system [13].
- To integrate new fault tolerance techniques with existing workflow scheduling algorithms the new approach needs to be developed [14].
- There is a requirement to implement autonomic fault tolerance technique for multiple instances of an application running on several virtual machines [12].
- To ensure high reliability and availability multiple clouds computing providers with independent software stacks should be used [15] [17].
- For evaluating the performances of fault tolerance component in comparison with similar ones a benchmark based method can be developed in cloud environment [8].

According to various researchers, two main factors which affect the performance of cloud computing are:

A. Dynamic Scalability

As the number of user requests increase the application must be able to support the increasing load. At the same time as the number of requests decrease the application should be able to scale down. So achieving dynamic scalability is a challenge in the cloud. Windows Azure randomly de-allocates the compute nodes when scaling down and hence follows an asynchronous process. This negatively impacts the performance. Both IaaS and PaaS provide services may be used to build and deploy scalable applications that can be optimized for parallel computing.

B. Fault Tolerance

Virtual nodes are created on demand to handle the load and to perform the computing tasks. So providing Virtual resources is another challenge for an application or HPC application in the cloud. If a virtual node fails while performing a task it becomes imperative to identify where in the system it failed and why? In order for the performance to be not impacted it is important that the load is transferred to another node while it is identified and fixed. It is also referred to as fault tolerance. This is one of the challenges while designing load balancing systems for high performance applications. It is possible that a compute node may share its resources in running more than one application. As the number of applications increases the load on the compute node, it can decrease the performance and may sometime fail if reached over capacity. The concept of virtualization plays a key role for the implementation. Besides the construction of virtual on demand clusters there are further benefits that could be expected, such as live migration.

3. EXISTING SYSTEM

Author [1] has analyzed the application of byzantine fault tolerance to federated clouds and presented experimentation performed to analyses the effectiveness of Byzantine fault tolerance in federated Clouds. Author has developed a Cloud framework called FT-FC that allows us to very quickly create diversity-based Byzantine fault-tolerant systems and apply them to federated Clouds.

Author [2] has proposed a middle layer, its purpose is to tolerate node failure and it placed between application and virtualization layer in cloud architecture.
In cloud environment, author [3] has discussed the existing fault tolerance techniques their policies, tools used and research challenges. He also has been proposed virtualized system architecture based on HAproxy and in this proposed system autonomic fault tolerance has been implemented.
Various fault tolerance techniques along with some existing model and comparison between them are discussed by author [4].
Author [5] has proposed the framework for creating and managing fault tolerance in Cloud computing environment. This framework overcomes the inflexibility of application developers by shading the implementation details of the reliability techniques and offering the desired level of fault tolerance support as an on-demand service.
Author [6] has proposed the fault tolerance model for real time cloud computing. According to this model system tolerates the faults and makes decision on the basis of reliability of the processing nodes i.e. virtual machines. According to this model, virtual machine reliability is adaptive, which changes after every computing cycle.
Author [7] discussed about cloud computing, highlighting its key concepts, state-of-the-art implementation as well as research challenges.
Author [8] focused on obstacles in cloud and discussed about opportunities for growth of cloud computing.
Author [9] has analyzed the concept of autonomic repair in the implementation of FT in cloud computing environment. Author used collaborative approach to solve FT in cloud computing. Collaborative approach constitutes an interesting tradeoff between exclusive management by provider and exclusive management by the consumer.
Author [10] discussed about current state of HPC applications in cloud computing.
In cloud environment, author [11] presented a survey of the current state of cloud computing.

4. PROPOSED WORK

A lot of work has been done in the area of fault tolerance on cloud computing. But there is lot of research room available in fault tolerance. In this paper a method is proposed that is Fault Detection and Tolerant System (FDTS).This scheme use heartbeat and gossip algorithm to detect whether the application is working smoothly or not. In case the application is detected to be down then the proposed work deploys an application recovery mechanism applied at the SaaS layer, which will try to start, recover from failure or will reinstall the application so that users can use the same smoothly with minimum downtime. Following steps are used in this proposed scheme:

Step 1: A Fault Detection and Tolerant System (FDTS) shall be created.

Step 2: A Web Application for User Management shall be created.

Step 3: A Cloud Environment shall be mapped using a Virutal Machine having web application running on it shall be created.

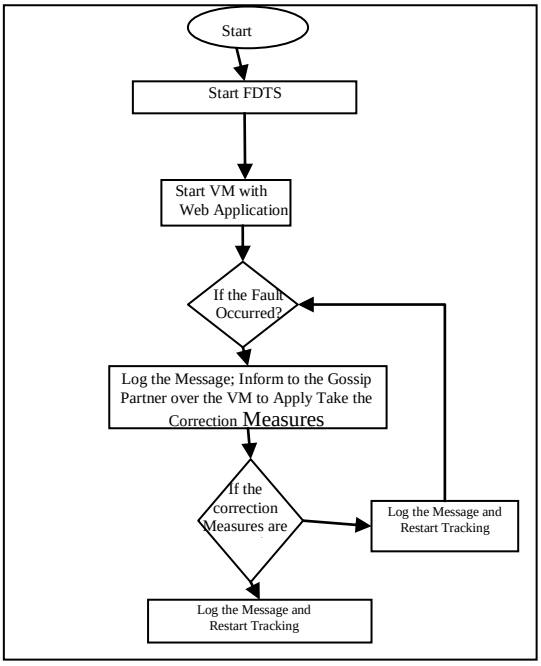
Sstep 4: FDTS will use heartbeat algorithm to detect if the web application is running continuously.

Step 5: If the FDTS loses the connection with web application it will test the web ports for the web application if they are working fine or not.

Step 6: The web application will be tried to be restarted using another application running over the virtual machine. For implementing this Gossip Protocol shall be used.

Step 7: All the generated shall be logged.

Flow chart for proposed scheme:-



5. CONCLUSION

Fault tolerance techniques are used to predict failures and take an appropriate action before failures actually occur. Failures are normal rather than exception in cloud computing environment, high fault tolerance issue is one of the major obstacles for opening up a new era of high serviceability. To achieve high level of cloud serviceability, a fault tolerance is needed. This paper discusses a technique for creating and managing fault tolerance in cloud. This technique uses heartbeat algorithm(s) and gossip algorithms to detect whether the application is working smoothly or not.

Our future work will be the implementation of framework to realize the proposed method to detect and remove fault tolerance.

In future we can also use some more algorithms in this scheme to enhance it so that our scheme should be more fault-tolerant.

REFERENCES

[1]Garraghan, P,Townend, P and Xu "J Byzantine fault-tolerance in federated cloud computing" 978-1-4673-0411-5, IEEE 2011.

[2]L.Arockiam, Geo Francis E "FTM- A Middle Layer Architecture for Fault Tolerance in Cloud Computing", 2012.
[3]Thakur Kapil Singh, Godavarthi Tarakarama RaviTeja,, Padmavathi Srinivasa Pappala " Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing" International Journal of Scientific and Research Publications, Vol 3, Issue 6, June 2013.
[4] Prasenjit Kumar Patra, Harshpreet Singh, Gurpreet Singh "Fault Tolerance Techniques and Comparative Implementation in Cloud Computing" Vol 64, feb 2013.
[5]R. Jhawar, V. Piuri, and M. D. Santambrogio "A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing" IEEE, Mar. 2012.
[6]Sheheryar Malik and Fabrice Huet "Adaptive Fault Tolerance in Real Time Cloud Computing" 2011 IEEE World Congress on Services.
[7]Qi Zhang, Lu Cheng, Raouf Boutaba "Cloud Computing: State-of-the-art and Research challenges" J Internet Serv Appl (2010) 1:7-18 DOI 10.1007/s 13174-010-0007-6
[8]Michael Armbrust, Armando Fox, Rean Griffith "Above the Clouds: A Berkeley View of Cloud Computing" Electrical Engineering and Computer Sciences University of California at Berkeley, 2009.
[9]Alain Tchana, Laurent Broto and Danial Hagimon "Approaches to Cloud Computing Fault Tolerance" 978-1-4673-1550-0/12,2012 IEEE.
[10]Sanjay P. Ahuja, Sindhu Mani "The State of High Performance Computing in the cloud" Journal of Emerging Trends in Computing and Information Science ,Vol. 3,Feb 2012.
[11]Harjit Singh "Current Trends in Cloud Computing A Survey of Cloud Computing Systems" International Journal of Electronics and Computer Science engineering, ISSN 2277-1956/V1N3-1214-1219
[12] http://haproxy.1wt.eu/download/1.3/doc/configuration.txt.
[13]Yang Zhang1, Anirban Mandal2, Charles Koelbel1 and Keith cooper," Combined Fault Tolerance and Scheduling Techniques for workflow Applications on Computational Grids "in 9th IEEE/ACM international symposium on clustering and grid, 2010.
[14]Imad M. Abbadi "Self-Managed Services Conceptual Model in Trustworthy Clouds' Infrastructure", 2010.
[15]Wenbing Zhao, P. M. Melliar-Smith and L. E. Moser," Fault Tolerance Middleware for Cloud Computing", 2010 IEEE 3rd International conference on cloud computing.