Volume 5, Issue 1, January 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Fault Tolerance in Cloud Computing by Efficient Load Balancing Mechanism Based on Ant Colony

Sonal Rana
CSE/PTU
Kapurthla, India

Aditi Sharma
CSE/PTU
Kapurthla, India

Amandeep Kaur
CSE/ Punjab Tech. Board
Chandigarh, India

Abstract: Cloud computing provide many service to the user. Now a day the user level is highly increased to utilize the services in cloud computing. From a technical point of view, most cloud computing platforms exploit virtualization, which implies that they are split into 3 layers: hosts, virtual machines and applications. This structuring of cloud makes it difficult to implement effective management policies .In cloud computing the major problem area is fault tolerance. Fault tolerance is a major concern to guarantee availability and reliability of critical services as well as application execution. In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. The load balancing should be a good fault-tolerant technique. This paper discusses about the load balancing algorithms which supports fault tolerance for the cloud.

Keywords: cloud computing, Load balancing, fault tolerance, Ant Colony.

I. INTRODUCTION

Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards. It is distinguished by the notion that resources are virtual and limitless and that details of the physical systems on which
software runs are abstracted from the user. In an effort to better describe cloud computing, a number of cloud types have been defined. Cloud is divided into two classes:

- One those based on the deployment model and
- One those based on the service model.

The deployment model tells you where the cloud is located and for what purpose. Public, private, community, and hybrid clouds are deployment models. Service models describe the type of service that the service provider is offering. The best-known service models are Software as a Service, Platform as a Service, and Infrastructure as a Service—the SPI model. The service models build on one another and define what a vendor must manage and what the client's responsibility is. Cloud computing takes the technology, services, and applications that are similar to those on the Internet and turns them into a self-service utility. The use of the word "cloud" makes reference to the two essential concepts:

• **Abstraction:** Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren't specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous.

• **Virtualization**: Cloud computing virtualizes systems by pooling and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility

II. TAXONOMY OF CLOUD COMPUTING

2.1 Cloud Architecture

Fig.1 shows the cloud architecture .Cloud architecture is the design of software application that uses internet accessible on demand services.

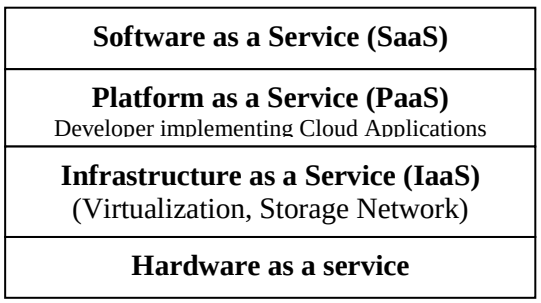


Fig.1 Cloud Architecture

According the NIST definition of cloud computing there are two types of cloud :

2.2 Deployment model-this refer to location and management of the cloud infrastructure.

These are four types:

- Public Cloud
- Private Cloud
- Community Cloud
- Hybrid Cloud

2.3 Service model-this refer to particular types of services that you can access on a cloud computing platform. These are three types

- IAAS
- PAAS
- SAAS

2.3.1 Infrastructure as a Service: IaaS refers to on-demand provisioning of infrastructural resources, usually in terms of VMs. The cloud owner who offers IaaS is called an IaaS provider. Examples of IaaS providers include Amazon EC2 , Go Grid and Flexi scale.

2.3.2 Platform as a Service: PaaS refers to providing platform layer resources, including operating system support and software development frameworks. Examples of PaaS providers include Google App Engine, Microsoft Windows Azure

2.3.3 Software as a Service: SaaS refers to providing on demands applications over the Internet. Examples of SaaS providers include Salesforce.com Rackspace and SAP Business In particular, five essential elements of cloud computing is clearly articulated.

2.4 On-demand self-service: A consumer with an instantaneous need at a particular timeslot can avail computing resources (such as CPU time, network storage, software use, and so forth) in an automatic (i.e. convenient, self-serve) fashion without resorting to human interactions with providers of these resources.

2.5 Broad network access: These computing resources are delivered over the network (e.g. Internet) and used by various client applications with heterogeneous platforms
(Such as mobile phones, laptops, and PDAs) situated at a consumer's site.

2.6 Resource pooling-A cloud service provider's computing resources are 'pooled' together in an effort to serve multiple consumers using either the multi-tenancy or the virtualization model, "with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

2.7 Rapid elasticity: For consumers, computing resources become immediate rather than persistent: there are no up-front commitment and contract as they can use them to scale up whenever they want, and release them once they finish scaling down. Moreover, resources provisioning appears to be infinite to them, the consumption can rapidly rise in order to meet peak requirement at any time.

2.8 Measured Service-Although computing resources are pooled and shared by multiple consumers (i.e. multi-tenancy), the cloud infrastructure is able to use appropriate mechanisms

III. FAULT TOLERANCE TECHNIQUES IN CLOUD

There are various techniques available to provide fault tolerance. Below given techniques achieve fault tolerance by load balancing.

1) **Replication**-Various task replicas are run on different resources, for the execution to succeed till the entire replicated task is not crashed. It can be implemented using tools like HAProxy, Hadoop and AmazonEc2 etc.
2) **Job Migration**-During failure of any task, it can be migrated to another machine. This technique can be implemented by using HAProxy.
3) **SGuard**-It is less disruptive to normal stream processing and makes more resources available. SGuard is based on rollback recovery and can be implemented in HADOOP, Amazon EC2.
4) **Self-Healing**- When multiple instances of an application are running on multiple virtual machines, it automatically handles failure of application instances.

There are various faults which can occur in cloud computing .Based on fault tolerance policies various fault tolerance techniques can be used that can either be task level or workflow level.

3.1 Reactive fault tolerance: Reactive fault tolerance policies reduce the effect of failures on application execution when the failure effectively occurs. There are various techniques which are based on these policies like Checkpoint/Restart, Replay and Retry and so on.

- **Check pointing/ Restart -** When a task fails, it is allowed to be restarted from the recently checked pointed state rather than from the beginning. It is an efficient task level fault tolerance technique for long running applications.
- **Replication**-Various task replicas are run on different resources, for the execution to succeed till the entire replicated task is not crashed. It can be implemented using tools like HAProxy,Hadoop and AmazonEc2 etc.
- **Job Migration**-During failure of any task, it can be migrated to another machine. This technique can be implemented by using HAProxy.
- **SGuard**- It is less disruptive to normal stream processing and makes more resources available. SGuard is based on rollback recovery and can be implemented in HADOOP, Amazon EC2.

- **Retry**-It is the simplest task level technique that retries the failed task on the same cloud resource.
- **Task Resubmission**-It is the most widely used fault tolerance technique in current scientific workflow systems .Whenever a failed task is detected, it is resubmitted either to the same or to a different resource at runtime.
- **User defined exception handling**-In this user specifies the particular treatment of a task failure for workflows.
- **Rescue workflow**-This technique allows the workflow to continue even if the task fails until it becomes impossible to move forward without catering the failed task.

3.2 Proactive Fault Tolerance

The principle of proactive fault tolerance policies is to avoid recovery from faults, errors and failures by predicting them and proactively replace the suspected components other working components. Some of the techniques which are based on these policies are Preemptive migration, Software Rejuvenation etc.

- Software Rejuvenation-It is a technique that designs the system for periodic reboots. It restarts the system with clean state.

3.3 Challenges of Implementing Fault Tolerance in Cloud Computing

Providing fault tolerance requires careful consideration and analysis because of their Complexity, inter-dependability and the following reasons.

- There is a need to implement autonomic fault tolerance technique for multiple instances of an application running on several virtual machines.
- Different technologies from competing vendors of cloud infrastructure need to be integrated for establishing a reliable system.
- The new approach needs to be developed that integrate these fault tolerance techniques with existing workflow scheduling algorithms.
- A benchmark based method can be developed in cloud environment for evaluating the performances of fault tolerance component in comparison with similar ones.
- To ensure high reliability and availability multiple clouds computing providers with independent software stacks should be used.
- Autonomic fault tolerance must react to synchronization among various clouds.

IV. IMPORTANCE OF LOAD BALANCING

Load balancing is one of the important factors to heighten the working performance of the cloud service provider. Balancing the load of virtual machines uniformly means
that anyone of the available machine is not idle or partially loaded while others are heavily loaded. One of the crucial issue of cloud computing is to divide the workload dynamically.

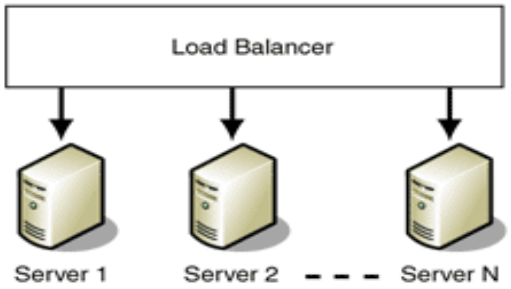


Figure 2- Load Balancing

If this issue is not addressed then the users have to face the problem of timeouts, response delays and long processing time.

4.1 Goals of Load Balancing

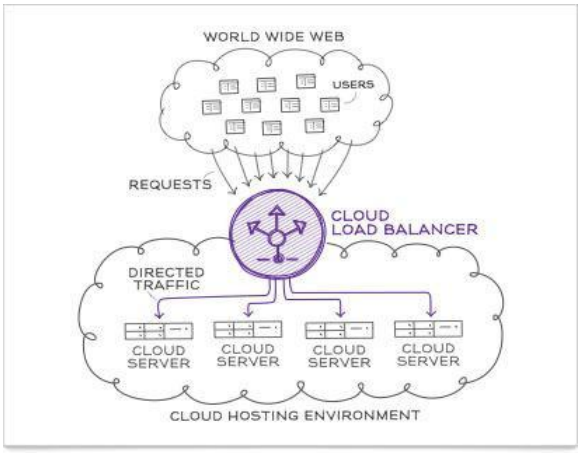
Goals of load balancing as discussed by authors of include:

- Substantial improvement in performance
- Stability maintenance of the system
- Increase flexibility of the system so as to adapt to the modifications.
- Build a fault tolerant system by creating backups.

4.2 Load Balancing Algorithms

In this section we have discussed some of the load balancing algorithms which helps to build a fault tolerant system.

1. Load Balancing strategy for Virtual Storage
2. ACCLB (Load Balancing mechanism based on ant colony and complex network theory)
3. Central Manager Algorithm
4. Central Queue Algorithm
5. Local Queue Algorithm



4.3 ACCLB (Load Balancing mechanism based on ant colony and complex network theory)

Z. Zhang etal. Proposed a load balancing mechanism based on ant colony and complex network theory in an open cloud computing federation. It uses small-world and scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excellent in fault tolerance and has good scalability hence helps in improving the performance of the system. In this algorithm, A central processor selects the hostfor new process. The minimally loaded processor depending on the overall load is selected when process is created. Load manager selects hosts for new processes so that the processor load confirms to same level as much as possible. From then on hand information on the system load state central load manager makes the load balancing judgment. This information is updated by remote processors, which send a message each time the load on them changes. This information can depend on waiting of parent's process of completion of its children's process, end of parallel execution. The load manager makes load balancing decisions based on the system load information, allowing the best decision when the process created. High degree of

Inter-process communication could make the bottleneck state. This algorithm is expected to perform better than the parallel applications, especially when dynamic activities are created by different hosts.

4.4 Performance of Load balancing Algorithm

The performance of load balancing algorithms is measured by the various parameters like overload rejection, fault tolerance, forecasting accuracy, stability, resource utilization etc., The load balancing algorithms promises to build fault tolerant systems, means that the algorithm is able to tolerate tortuous faults and continue operating properly in the event of some failure. If the performance of algorithm decreases, the decrease is proportional to the seriousness of the failure.

V. CONCLUSION

Fault tolerance is one of the main challenges in cloud computing. This paper identified load balancing algorithms based on ant colony theory 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. When all these issues are addressed naturally the system becomes a fault tolerant one.

REFERENCES

[1] Z. Zhang, and X. Zhang, "A Load BalancingMechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pages 240- 243.

[2] Zhang Z. and Zhang X. (2010) 2nd International Conference on Industrial Mechatronics and Automation, 240-243.

[3] William Leinberger, George Karypis, Vipin Kumar, "Load Balancing Across Near-Homogeneous Multi-Resource Servers", 0-7695-0556- 2/00, 2000 IEEE.

