Clustering for Self-Deployed Cloud Applications

Jobin Babu[1], Varun Sharma[2], Anil Saroliya[3]
[1]Student M.tech ,[2,3]Senior Lecturer Computer Science & Engineering
[1, 2, 3] Amity University, Rajasthan, India

Abstract— There are certain methods to create auto-scaling applications to make applications according to the load in cloud computing. Disaster recovery is one of the usages of this environment. For an example, during the event of disaster, applications may have the freedom to move from one environment to another one. Applications which are continuous in nature such as continuous integration tests require environments that support their running context. These applications are treated as individual applications for development. Clustering in the cloud computing can help in developing the applications as self-deployed applications. This paper describes the possibility of this type of phenomenon and helps us to discuss it through a fundamental structure.

Keywords: - Cloud computing; clustering; Self deployable application, Iaas.

I. INTRODUCTION

It is relatively easy for a particular application to use their own execution environment when running on a individual virtual machine. An example of support from one environment to another environment is the live migration mechanism. But, generally clusters require machine clusters for their smooth and sensible execution. Such an example is presented here which has three types of triggers for deployment.

A. Context

This type includes those test tools which are continuous in nature and which generate test environments on demand depending on the test configurations at that time.[3] Applications Requiring large amounts of memory on demand are also examples in this type.

B. Amount of transactions to applications

Web servers, database servers and application servers are the three tiers which is consisted in the auto scaling of web services. These examples are appropriate for this type.

C. Geographically constraint on resource for applications

Network latency can be in the area of sensitivity for applications such as online games. So, they have a requirement of execution close to the users. "Follow the Sun" use cases which are similar to that of on-line games can be specimens. Another case can consist of those applications which have the need to use geographically constrained resources such as tremendous amounts of information or special hardware. Applications have to be mobile to other locations where they can be run safely. This creates the necessity of disaster recovery.

The issues arising in preparing the environment of execution for different categories of deployment triggers can be studied. Cost of educating application developers is the main concern. Developers have to be educated for the solutions for each type. Development costs of cloud foundations can be the other concerned area. So, common parts which are present in the cloud foundation may not be redeveloped many times. Applications may belong to multiple types. Web services can be taken as a example. Here, we are taking two categories. Web services which belong to category a have to prepare for disasters. It means that web service developers require to add or provide solutions of type A and B at the same time which is not an easy task as these solutions are always been in habit of being individually designed.

In this paper, an idea is being presented called Clustering as a Service which will solve the issues. It can allocate resources for a computer and can implement applications on it accordance with the request from an application.[1]

II. NECESSITY

The above problems resulted to the creation of such an idea which solve all the above issues and it can be effectively done through clustering.[1] The idea has the potential to resolve all the issues. In Cluster as a service, service providers manage or administrate computer resources which are common and allocate those resources when demanded by any individual application.

There are some requirements which the foundation has to follow to provide the service well and to execute and yield accurate result.[6] They are discussed below properly. Requirements such as:

- The resources must be allocated in a dynamic manner to the clusters of the various applications strictly.
- Clustering should be done in a way such that those clsters created should be securely seprated by each other.
- Clusters should be created in such a way that components of the software can be easily implemented on the clusters.
- This service can also be called through a web API.

III. CLUSTERING

Here, cluster as a service is designed to meet all of the above requirements.[4] So, for the proper implementation of the clusters, it has been decided to divide it into two definite layers. It is explained below:

The two layer implementation has been done intentionally. There are specific duties assigned to both of the layers. The lower layer is there for the preparation of nodes while the upper layer is for the installation of software on nodes. These layers, on the other hand, also take care of the requirements such as the lower layer takes care of the first two requirements and the upper layer takes care of the third requirement. Both of the layers are the accessible with web API's.

The lower layer consist of the mechanism of using machine images for handling the operating system of the nodes which are composed in the cluster. Nodes can be

allocated dynamically with the help of software and securely can be separated with the help of network technology.[5]

The upper layer deals with the software components. The upper layer implements the software components of the application frameworks. Hadoop, sun grid engine and openstack are the examples of those frameworks.

IV. THE PROTOTYPE

Here, we are going to introduce an implementation in clustering called Dodai which helps in managing multiple clusters of different frameworks of different applications. So, users just have to provide their configuration and they will be provided by their clusters.[3] Dodai is composing of tool named as Dodai-deploy which can be described as the Dodai upper layer and Dodai-compute which is the lower layer of the cluster as a service.

The Dodai-deploy may have both graphical interface and command line as a convenience but the Dodai-compute only would have a command line interface. The Dodai- compute having a command line interface is designed so close to the interface of the virtual machines of big guns such as Amazon. Through this , application developers manage bare-metal machines.thus, this type of design results into a condition were it is easy for the applications to deploy themselves in accordance with their demand.

Now let us talk about the second most important part of the cluster as a service i.e the upper layer Dodai-deploy.[3] As discussed earlier, the job of Dodai-deploy is to handle the operating systems. Going to the much detailed part of it, Dodai-deploy is a simply a software implementation of the cluster ass a service. It supports Hadoop 0.20.2, Sun Grid Engine 6.2u5 and Openstack Diablo (Nova, Glance, and Swift).

The Dodai-deploy has some of the unique features which make it very essential. They are as follows:

- It deals with installation, un-installation, and testing management.
- It supports the target machines and it can be done through different network segments.
- It provides web UI's to users for their convenience to support user operations.
- It introduces an API named as REST API which has the quality of integration of a tool with other tools.
- It can install softwares in a parallel manner.
- It is a means to deploy target software.

The Dodai-deploy produces puppet clients in accordance with the demand produced and the MCollective manages the puppet clients on target nodes which is done in a parallel way so that deployment of softwares could be done at a quicker rate.

Last but not the least the Dodai-compute is tool which deals with the nodes for the upper layer. The nodes can be bare metal achiness or virtual machines.

V. CONCLUSION

So, the prototype described here can be used as a successful practical implementation which satisfies the requirements for the self-deployable applications. Let us take an example of continuous integration environment which could be deployed on demand depending on the test configurations. A whole new private cloud structure could be created with the help of this architecture and it will help cloud in giving compatibility to different application frameworks. This can be used well in research.

REFERENCES

[1] Graeme Philipson, "Why is Cloud Computing important", http://www.itwire.com/20 12-06-0 1-13-40 03 /browse/c-leveIl55713 -why-cloud-computing-is important, july2012.
[2] OpenFlow: http://www.openflow.org
[3] Dodai: https://github.com/nii-cloud/dodai
[4] Puppet: http://puppetlabs.com
[5] MCollective: http://puppetlabs.com/mcollectiv
[6] Carlos Escapal, "Disaster recovery is ripe for cloud disruption", http://gigaom. Com/cloud/disaster-recovery-is ripe-for-cloud-disruption, February 2012.

