ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

Automated Scaling of Cluster Using Deployment Diagrams in Platform-As-A-Service

Sudhir S. Kanade*, Pushkaraj B. Thorat

HOD, Department of ENTC, COE, Osmanabad, India

ME, Department of Computer, COE, Osmanabad, India

ABSTRACT*:* Cloud computing is delivery of computational power in the form of service. In here software, computational resources and data are shared to computers and other devices as a metered service over the network. Platform-as-a-Service is a one of the cloud based approach that provides managed middleware, isolating user from managing resources and platform. In this paper we propose a system which uses Deployment Diagrams to manage applications and its whole cluster including middleware. Here we emphasis on its capablity to add/remove resources to the cluster, i.e. when demand is increased new nodes are created and plugged into the cluster depending on the deployment diagram grammar, and when the demand is lessen the resources are claimed and returned to the free pool.

KEYWORDS*:* cloud; IaaS; PaaS; SaaS; High availability cluster; Scalability

I. Introduction

Cloud computing is delivery of computational power in the form of service. Upon request the resources are allocated to the requesting entity from the shared pool and when the work is done they are relinquished and the system further adds it back to resource pool. Here resources such as network, storage, server in form of compute, applications complying SOA, and services that can be frequently provisioned and released with less management effort or less interaction of service provider [1].

The level at which the computational services are delivered further classifies it into:
1. Software as a Service (SaaS). The capability provided to the consumer is to use the provider applications running on a cloud infrastructure. Various client devices can access the application from various client devices through either a thin client interface from a web browser or through a program interface.
2. Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure created or acquired applications created using programming language, services, libraries and tools supported by the provider.
3. Infrastructure as a Service (IaaS). Infrastructure as a Service (IaaS), provides bare hardware as a service. It contains abstracts hardware entities like CPUs (power & cores), memory, network, storage exported via block and file, network bandwidth and connectivity. This form of service is rudimentary and primitive, this acts as foundation for PaaS and SaaS layers, which provides a flexible, usable and standard form of computing resource.

Because of flexibility of handling resources in cloud computing environment, application deployed on it has lot of advantages, important for us in this paper is - High Availability and Scalability[2]. The capability of cloud enabled application could be increased quickly to address the bullish user base, and same could be returned to the resource pool when the need is relinquished. In this discussion we are highlighting scale-in/out type of scalability under discussion. But the same concept could be applied to scale-up/down behaviour of sclability too.

High-availability and scalability has to honour the rules of cluster on which the application is deployed. To capture the grammar of cluster we use Deployment Diagrams. Our proposed implementation uses PaaS service model and deployment diagrams to capture the HA cluster grammar and aims to address all the five essential characteristics of

10.15680/ijircce.2014.0212055

cloud namely - On demand self services, Broad network access, Resource pooling, Rapid elasticity, Measured service, Multi Tenacity[2]. In this paper could be implemented for all types of cloud deployment includes but not limited to Private, Public, Community and Hybrid Cloud.

II. RELATED WORK

Scalability is one of the key benefits of using cloud computing. Scalability is the ability of the system to cope up with the increase loads by adding resources and/or nodes while the cluster is active and running, without disrupting the active and ongoing active transactions. Depending on business demands cluster can be easily up scaled or downscaled. For example, Movie ticket booking applications are heavily used on weekends and holidays and not used working days, in such less usage time the servers can be returned back to resource pool and given to organizations in other time zones, or having heavy usage and in same time zone can use them. When business needs are changed then cloud service providers can increase existing resources without any need of expensive changes existing IT systems[3][4].

There are two types of scaling: Horizontal scaling and Vertical scaling. Horizontal scaling is used when it is not possible to change one resource type to other. If IaaS layer is unable to scale up a virtual machine at runtime due to host machine resource limitations or not supported by hyper-visor, PaaS has to choose Horizontal scaling viz scale out and scale in such cases. In which nodes(i.e. servers) are added(scale out) or removed(scale in) to the system as per requirement with less processor and RAM. In vertical scaling we add/remove(scale up/down) resources(processor and memory) in the system. It is easy to implement but more costly than vertical scaling.

Modern day applications requires support from middleware like database, web servers etc. To deploy and scale up/down, application at runtime we need to manage runtime deployment of middleware, initializing it and adding the server to the application cluster. Lack of such framework makes the legacy applications to fail to get deployed over cloud environment.

Similar work for Service Oriented Architecture (SOA) for deploying SOA based solutions on PaaS and presented by Bao Rong [3]. But here we are collaborating the Deployment model and scaling out / in of platform for generic cluster applications.

One more approach is proposed which provide high availability and high scalability for the enterprise resource planning. The WebSphere cloudburst is solution provided by the IBM for resource monitoring, designed to speed creation and deployment of application to cloud and virtual environment. This solution does not take the load on the individual server into account to scaling out/scaling in the cluster. Solution proposed here has the capability so that virtual nodes in the cluster can report back the health and load the system is going through, hence can be used for automated cluster deployment.

III. PROPOSED SYSTEM OF AUTOMATED SCALABILITY OF CLUSTER USING DEPLOYMENT DIAGRAMS (ASCDD)

In our proposed system provide framework for automated and on-demand scaling with event driven mechanism. Virtual node on the cluster report back regarding its health and load to the monitoring server. When load at the virtual node increases then the agent present at each node report back to the monitoring server about it, which then scale out the component of particular role. Similarly, if load at virtual node decreases then the agent present at each node report back to the monitoring server about it, so that it then scale in the component of particular role. Deployment diagram and scripts are fed to management server for configuration.

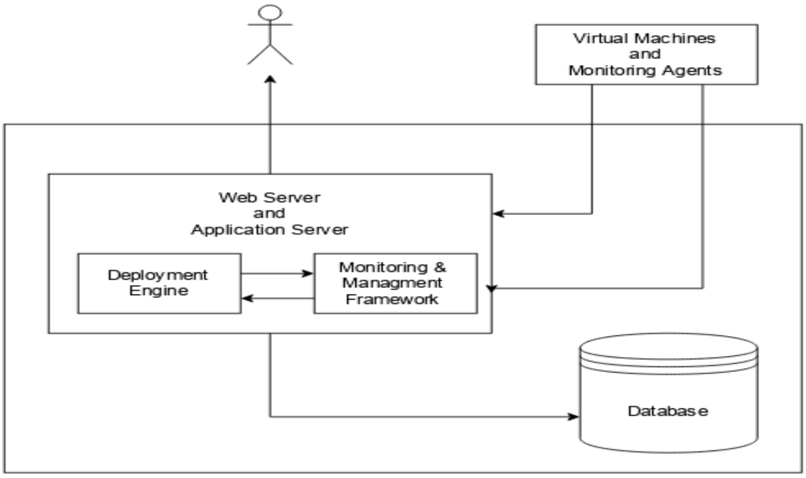

Fig. 1 Architecture of ASCCD

The system will contain following parts:

1. Deployment engine (DE)
2. Web UI
3. Monitoring/Managing framework(MMF)

(a) Monitoring agents

Deployment Engine (DE) is responsible for installing virtual machines images, getting them into network, installing monitoring agents, installing middle ware followed by configuring them. DE is not a self driven entity but it will be managed by MMF.

Monitoring and Managing Framework (MMF) reads topology file of deployment diagrams and create a base platform definition, this definition can be modified later through WEB UI. This platform definition will consist following information.

1. Architecture of a Platform including multiplicity of middleware and VM specifications.
2. Rules

Architecture of a Platform, defines, architecture required by application to run and Rules, to when to deploy an application and when to destroy it.

MMF will take help of plug-in related to different middleware for deploying / managing / destroying them.

Web UI will provide overall management of fabric, consisting:

i. Creating/Updating topology file.
ii. Testing topology file.
iii. Deploying a middleware platform.
iv. Getting status/health of platform.

Below proposed algorithm could be used for adding a node in a cluster:

```
addNode(role, isFreshDeploymentRole=False):
  IPAddress = getNextFreeIPAddress()
  host = getPreferredHost(role)
  scriptPath = composeRoleDeployFile(role, host, IPAddress)
  copy script file to host machine
  execute script from host machine
```

10.15680/ijircce.2014.0212055

```
  until (vm is started)
    sleep(1);
  copy and execute role configure script to vm.

  if isFreshDeploymentRole:
    for roleDep depending on role
      for avirtualmachine of roleDep
        Notify avirtualmachine that new node is added
  else:
    for all roles depending on role 'role'
      for avirtualmachine of roleDep
        Notify avirtualmachine that new node is added
```

Below algorithm was used for removing the node from cluster:

```
removeMachine(clusterVM):
  for roleDep dependent in clusterVM.RoleName:
    for avirtualmachine of role roleDep.RoleName:
      Notify avirtualmachine that clusterVM is being removed

    unconfigure avirtualmachine
    stop middleware on avirtualmachine
```

In the test environment we experimented on the above system and found that the architecture is linear capable of monitoring and delivering nodes in the cluster as per the load varies.

IV. Experimental Results

In our lab we experimented on the above system and found that the architecture is capable of monitoring and delivering nodes in the cluster as per the load varies. We could inject a node of a role in a cluster with around 1 node/min.The results are very astounding thought there is plenty of room for improvement. Below are the operations which were experimented:

1. Increasing load on cluster
2. Decreasing load on cluster

Increasing load on cluster

When the load on system is increased with rate of 1000 requests/min, when per node capability of serving 100 requests we could scale up in O(n) linear scalability.

Decreasing load on cluster

When the load on system is decreased with rate of 1000 requests/min, when per node capability of serving 100 requests we could scale down in O(1) constant time.

V. Conclusions

This paper introduces the platform that provides an effective way to share resources on cloud and automated and on-demand scalability with event driven management. The resulting system could be more durable to load and stress over the resources, without need of manual intervention. The algorithms could be improved further to analyze the patterns when the system is under load and take corrective action before hand.

10.15680/ijircce.2014.0212055

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2014

REFERENCES

[1] Hailong Sun, Xu Wang, Minzhi Yan, Yu Tang, Xudong Liu "Towards a Scalable PaaS for Service Oriented Software", 2013 IEEE International Conference on Parallel and Distributed Systems

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, U. S. Department of Commerce, 2011.

[3] Bao Rong Chang, Hsiu-Fen Tsai, Ju-Chun Cheng, Yun-Che Tsai High Availability and High Scalability to in-Cloud Enterprise Resource Planning System", Intelligent Data analysis and its Applications, Volume II Advances in Intelligent Systems and Computing Volume 298, 2014.

[4] Bin Claudio A. Ardagna, Ernesto Damiani, Fulvio Frati, Davide Rebeccani, "Scalability Patterns for Platform-as-a-Service", 2012 IEEE Fifth International Conference on Cloud Computing.

[5] X. Suo, Y. Zhu, and G. S. Owen, "CLOUD RESOURCE PROVISIONING AND BURSTING APPROACHES", 2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing.

[6] ZENG Shu-Qing "The Improvement of PaaS Platform" 2010 First International Conference on Networking and Distributed Computing

[7] Srijith K. Nair, Sakshi Porwal, Theo Dimitrakos, Ana Juan Ferrer, Johan Tordsson, Tabassum

[8] Sharif, Craig Sheridan, Muttukrishnan Rajarajan and Afnan Ullah Khan., "Towards Secure Cloud Bursting, Brokerage and Aggregation", 2010 Eighth IEEE European Conference on Web Services

[9] Nishant Agnihotri1,Aman Kumar Sharma "EVALUATING PAAS SCALABILITY AND IMPROVING PERFORMANCE USING SCALABILITY IMPROVEMENT SYSTEMS", IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163 I. Llorente, OpenNebula Project. http://www.opennebula.org/

[10] Amazon Elastic Compute Cloud (EC2), http://www.amazon.com/ec2/ Google App Engine, http://appengine.google.com

[11] Sun network.com (Sun Grid), http://www.network.com

[12] The NIST Definition of Cloud Computing http://csrc.nist.gov/publications/nistpubs/800-145/SP800.pdf

[13] Where are we at with Cloud Computing?: A Descriptive Literature Review – Yang, Tate http://www.infotech.monash.edu.au/about/news/conferences/acis09/Proceedings/pdf/079.pdf

[14] Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities http://arxiv.org/ftp/arxiv/papers/0808/0808.3558.pdf

10.15680/ijircce.2014.0212055