Enhance Load Balancer Behavior Identifier for Dynamic Auto-Scaling in Cloud

Rupal Gohel[1], Gaytri Pandi[2]

[1]P.G. Student, [2]Assistance Professor & Head

[1,2]Department of Computer Engineering

[1,2]L.J. Institute of Engineering and Technology, Ahmedabad, Gujarat, India

Abstract

Cloud computing is a technology to provide resources from the large data centers. Cloud computing is made available as a services to the users. Rackspace, Salesforce, Amazon, Google, IBM, Dell and HP are the well-known service providers. It is a pay as you go model. In order to service providers have to improve the scalability, resource utilization and the providers use the auto-scaling mechanism to scale the resources according to the users need. In this paper we create a rule and find the dynamic threshold value for reduce the resource utilization and performance work fast.

Keywords—*Cloud Computing, Auto-Scaling, Virtual Machines, Load Balancer*

I. INTRODUCTION

Cloud computing is an emerging computing model. Cloud computing is the most widely used platform now a days. The US National Institute of Standards and Technology(NIST)[8] defines cloud computing as follows: " cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with a minimal management effort or service provider interaction.

This cloud model promotes availability and is composed of characteristics like on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services. And three service models like infrastructure as a service(IAAS), platform as a service (PAAS), software as a service(SAAS), four deployment models(public, private, hybrid, community).

Cloud computing in Auto Scaling is the ability to scale up or down the capacity automatically according to conditions of the user define. With Auto Scaling ensure that the number of instances is increasing seamlessly during demand spikes to maintain performance, and decreases automatically during demand reduce to minimize costs. The auto scaling in cloud infrastructure is shown in Fig.1

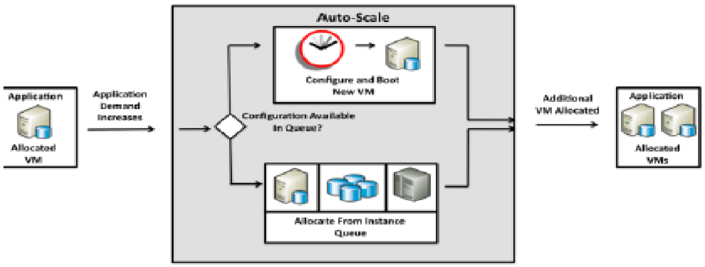
Application
Allocated VM
Application Demand Increases
Auto-Scale
Configure and Boot New VM
Configuration Available In Queue?
Allocate From Instance Queue
Additional VM Allocated
Application
Allocated VMs

Fig.1 auto-scaling in cloud infrastructure

Benefits of Auto-Scaling:

Better fault tolerance: Auto Scaling can detect when an instance is unhealthy, terminate it, and launch an instance to replace it.

Better availability: They can configure Auto Scaling to use multiple Availability Zones. If one Availability Zone becomes unavailable, Auto Scaling can launch instances in another one to compensate.

Better cost management: Auto Scaling can dynamically increase and decreas e capacity as needed, pay for the EC2 instances use, save money by launching instances when they are actually needed and terminating them when they aren't needed.

II. SCALABILTY IN CLOUD COMPUTING

Cloud scalability has two dimensions, namely horizontal cloud scalability and vertical cloud scalability.

Horizontal Cloud Scalability:

Horizontal cloud scalability is the ability to connect multiple hardware or software entities, such as servers, so that they work as a single logical unit. It means adding more individual units of resource doing the same job. In the case of servers, you could increase the speed or availability of the logical unit by adding more servers. Instead of one server, one can have two, ten, or more of the same server doing the same work. Horizontal scalability is also referred to as scaling out, which is shown in Fig. 2.

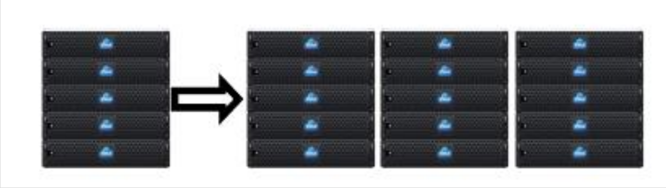

Fig. 2. Horizontal Scalability

Vertical cloud scalability:

Vertical scalability is the ability to increase the capacity of existing hardware or software by adding more resources. For example, adding processing power to a server to make it faster. It can be achieved through the addition of extra hardware such as hard drives, servers, CPU‟s, etc. Vertical scalability provides more shared resources for the operating system and applications. Vertical scalability may also be referred to as scaling up, which is shown in Fig.3.

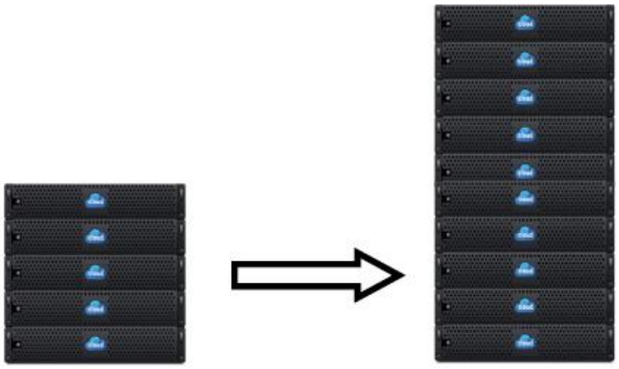

Fig. 3. Vertical Scalability

III. RELATED WORK

A. Implementing a Novel Load-aware Auto Scale Scheme for Private Cloud Resource Management Platform.

Auto Scale allows users to scale their cloud resources capacity according to elastic loads timely, which has been widely used in mature public cloud. For private cloud, there are some different features from public cloud. It is more flexible to use Auto Scale technique to provide QoS guarantees and ensure system health. .

Auto Scale allows users to scale their cloud resources capacity according to elastic loads timely, which has been widely used in mature public cloud. For private cloud, there are some different features from public cloud.[1]
Design a novel Auto Load-aware Scale scheme for private cloud environment in which scale in and scale out strategy based on prediction algorithm.

B. Cloud Resource Auto-scaling System based on Hidden Markov Model (HMM)

Elasticity characteristic of cloud computing enables users to acquire and release resources on demand, which reduces their cost by making them pay for the resources they actually have used. Proposed an auto-scaling system based on Hidden Markov Model (HMM).
HMM is a rich mathematical structure which can form a theoretical basis for use in a wide range of applications.
The Hidden Markov Models are tool for time series data modeling. They are used in almost all current speech recognition systems, applications in computational molecular biology, data compression, artificial intelligence and pattern recognition. In other words, HMM is a tool for representing probability distributions over sequences of observations.
An HMM is characterized by five elements:
N: the number of states in the model
M: the number of observation symbols per state. The observation symbols represent physical output of the system being modeled.
A: probability distribution of state transition.
B: probability distribution of observation symbol in state *j*
the initial state distribution.

π: A complete specification of a HMM requires specification of two model parameters (*N* and *M*), specification of observation symbols, and the specification of three probability measures *A*, *B*.
This is used to indicate the complete parameter set of the model.

C. Measuring Prediction Sensitivity of a Cloud Auto-scaling System

Elasticity is key benefits of cloud computing which helps customers reduce the cost. Although elasticity is beneficiary in terms of cost, obligation of maintaining Service Level –Agreements leads to necessity in dealing with the cost -performance trade-off. Proactive auto-scaling is an efficient approach to overcome this problem. In this approach scaling actions are generated based on prediction results. They have been focusing on improving prediction accuracy in order to improve the efficiency of Auto-Scaling mechanisms. However, the sensitivity of auto-scaling mechanisms to the prediction results is neglected in the domain[3].

Then investigated the sensitivity of Auto-scaling mechanisms to the prediction results by evaluating the influence of performance predictions accuracy on the Auto-scaling actions. They compared actions of threshold based scaling techniques which are generated based on Support Vector Machine (SVM) and Neural Networks (NN) predictions. Our experimental results show that SVM is more accurate than NN, scaling decisions made by the two algorithms are identical in 91.5% of the time. Furthermore, they have shown that the optimal training duration for SVM and NN is about 60% of experiment duration. Support Vector Machine (SVM) and Neural Networks (NN) are the most effective algorithms to predict future system characteristics. Hence, in this work they measured the influence of these two machine learning algorithms (i.e., SVM and NN) on the scaling decisions made by threshold based technique.

D. VM Auto-Scaling in Hybrid Cloud for Scientific applications

Auto-scaling method to provide efficient resource utilization in a hybrid cloud computing environment. Tasks in Bag-of-Tasks (BoT) can run in parallel while tasks in workflow can be executed in the order of dependency. However, the proposed auto-scaling algorithm limited to specific Bag-of-Tasks in aerodynamics and workflows in protein annotation workflow. Then need an Auto-scaling method in order to perform applications in a general form of workflows. The Auto-scaling Algorithm is extend to consider Only tasks in Bag-of-Task, to support workflow as well. Initial scheduling schedules tasks to prevent waste of VMs within a deadline. Auto-scaling method can perceive delay and deadline violation to comparing actual start time and estimated start time of running tasks during monitoring interval[5].

A workflow is commonly represented by a directed acyclic graph (DAG). In a workflow, tasks have their own order, that is child tasks, can execute when parent tasks are finished. Tasks which have a workflow pattern are important to consider dependency and their order during the auto-scaling method. In experiment our auto-scaling method to prove our Workflow Scheduling can allocate VM to various kinds of workflow patterns. Then develop random workflow generation in order to apply a various workflow patterns. Then generate various patterns of workflow by using the random number of depth and the random number of parent tasks which represent dependency. And we also make the random number of tasks at each level.

E. Load Balancer Behavior Identifier (LoBBI) for Dynamic Threshold Based Auto-scaling in Cloud

Cloud computing services available as on demand self -service basis at anywhere, anytime with Pay-as-you-go model. It is one of the fastest growing technology. Cloud Service Provider's (CSP) are capable enough to provide the services at any point of time. Cloud Users (CU) are satisfied if they get the cloud service in affordable price. Scalability is the key technique to trigger the scaling as per the users request. Auto-scaling and virtualization helps to achieve cost effective scaling. Setting dynamic threshold values in a cloud environment should utilize the resources completely and prevents the physical server damage. It manipulates the CSP to accommodate more user in a physical server and also reduces the cost of the service. Then elaborates to set a dynamic threshold value in the physical server, load balancer behavior identifier mechanism is proposed to generate the rule and provide the resources dynamically.

```
Begin

    Predict the initial work load and boot VMs;
```

identifier the behavior of the LB

Find out the Workload $I = \int_{a}^{b} \left(\frac{a1/b1}{2}\right) * T$

where I = Current workload

a = upper limit of threshold value

b = lower limit of threshold value

a1 = available resources

b1 = number of user requests

T = time interval

CPU = ck/n

Ck = cpu utilization

n = number of nodes

Memory = Mem Used / Total mem

Network = Nk Used / Total network

```
Set LT = 20% , MT = 70% , UT = 80%;
 if ( I >= MT )
Check CPY        where CPY= Capability of VM
 if (CPY = true )
            Manage all the instances
else
             Dynamically add VM instances
       else if ( I > UT )
             Send all request to the new VM
       else if ( I < LT )
              Remove VM
       else
              manage the request with available VM
       End
```

IV. PROPOSED WORK

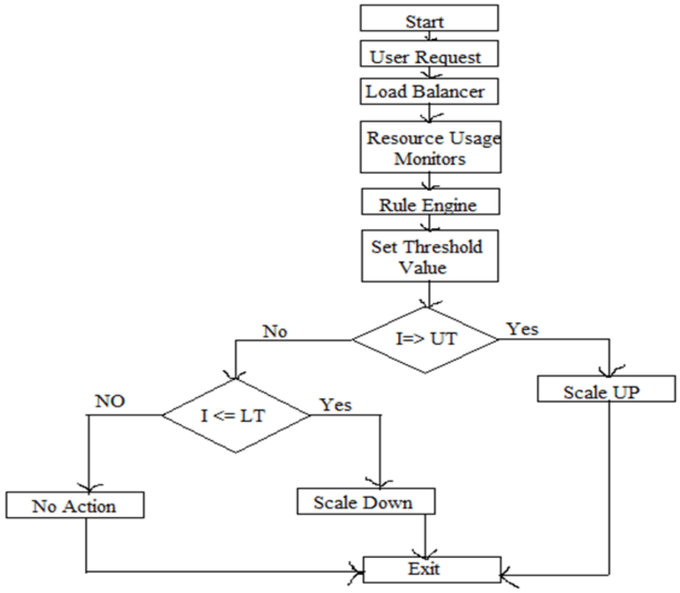

Fig.4 Logical Flow

Where I = Current workload

LT= Lower Threshold

UT= Upper Threshold

Use a Three parameters CPU, Memory and Network Throughput

In this proposed work,

Step1. Agent request is sent. It contains the end user data.

Step2. Load balancing is a method for distributing workload across multiple computing resources, such a computer, a computer cluster, network links, central processing units or disk drives. The load balancer collects the user request, then forwards the request to suitable VM according to the load balancing strategy.

Step3. It keeps on monitoring the load in a load balancer. finds out the number of CU request is in load balancer and number request served at a particular point of time. The behaviour of the LB keeps on changing with respect to the load conditions. It segregates the type of load into small, medium and heavy.

Step4. Rule engines activity is to receive the updated information. create a our rule and make a dynamic algorithm to find out the threshold value.

```
Procedure Rule Engine()
    For time ti=1 to n
        For each Vm's
            $DB_{cache} \rightarrow$ <$Vm_i$ , Utilfactor X>

                          Where A / alpha = CPU  ↓
                            B/ Bita = memory     ↓  X

                        // store information in DB Cache

  $min_i \rightarrow$ get_min_UtilFactor(DBCache)
  $max_i \rightarrow$ get_max_UtilFactor(DBCache)

    End For

  down_threshold $\rightarrow$ get_AVG_($min_i$)
  up_threshold $\rightarrow$ get_AVG_($max_i$)
```

Dynamic threshold based Auto-Scaling algorithm.

Step5. Auto-Scaling mechanism works when the rule engine triggers the dynamic rule. Auto-Scaling is used to add VM instances and remove VM instances as per the number of users.

IV. SIMULATION AND RESULTS

Simulation is done using amazon web service. From the results generated, it can be seen that the algorithm is capable enough of continuously adding newly created instances when such requirement arises and vice versa. The percentage of requests served are also increased while simulation of the algorithm is carried out. After the comparison of resource utilization by the previous algorithm and proposed algorithm, proposed algorithm is reduce the time and performance work fast.

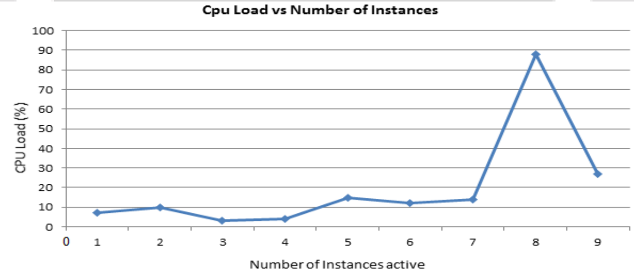

Fig.5 cpu utilization

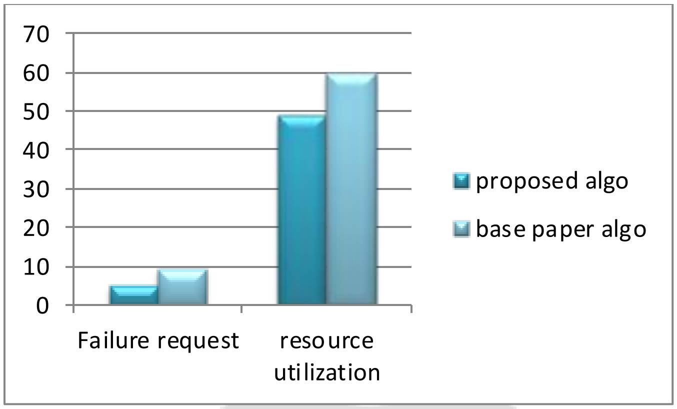

Fig.6 result comparison

V. CONCLUSION

Auto-scaling is one of the main challenges in cloud computing. In auto scaling we needed to know the present mechanism and the techniques used in auto scaling. Auto-scaling mainly focused to reduce the Cost, time saving, High availability and Fast Performance in work. The auto-scaling mechanism of the cloud system is the essential element in resource utilization, add or remove the infrastructure and management cost.
In future, to develop a this concept in large set of cloud data.

REFERANCES

1] Jie Bao , zhihui Lu , Jie wu , Shiyong Zhang , Yiping Zhong " Implementing a Novel Load-aware Auto scale Scheme for Private Cloud Resource Management Platform " school of computer science and Ministry of Education Shanghai,china,2014 IEEE,978-1-4799-0913-1/14,pp 240-247

[2] Ali Yadavar Nikravesh , Samuel A. Ajila " cloud Resource Auto-scaling system based on Hidden Markov Model (HMM),Department of systems and computer Engineering, CarletonUnivercity,2014IEEE,DOI10.1109/ICSC.2014.43,978-1-4799-4003-5/14,pp 124-127

[3] Ali Yadavar Nikravesh, Samuel A.Ajila "Measuring Prediction Sensitivity of a Cloud Auto-scaling System " Chung - Horng Lung Department of systems and computer Engineering, Catleton University ,2014 IEEE,DOI 1O.1109/COMPSACW.2014.116,978-1-4799-3578-9/14,pp 690-695

[4] Younsun Ahn ,Yoonhee kim "VM auto-scaling for workflows in hybrid cloud computing" Dept. of computer science sookmyung women's University seoul, Korea, 2014 IEEE, DOI 10.1109/ICCAC.2014.34,978-1-4799-5841-2/14315,237-240,pp 237-240

[5] M.Kriushanth , Dr.L. Arockiam "Load Balancer Behavior Identifier for Dynamic Threshold Based Auto-Scaling in cloud" Research Scholar in computer Science,St.Joseph's College, Tiruchirappalli, Tamil Nadu, India, 2015 ICCCI, 978-1-4799-6805-3/15,pp 150-156

[6] E. Barrett, E. Howley, J. Duggan, "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud," Cuncurrency and Computation: Practice and Experience, vol. 25, no. 12, pp 1656 – 1674, 2014