IJCSC 0973-7391 Volume 8 • Issue 2 March 2017 - Sept 2017 pp. 141-145 available online at www.csjournals.com

SLA Based an Efficient and Reliable Resource Provisioning in Private Cloud

Lata Gadhavi[1],Darpan Korat[2], Madhuri Bhavsar[3]
Department of Computer Science and Engineering
Institute of Technology,Nirma University
Ahmedabad,India
12extphde92@nirmauni.ac.in[1],darpankorat113@gmail.com[2],madhuri.bhavsar@nirmauni.ac.in[3]

Abstract: Cloud computing has turned into the most prevalent distributed computing environment in light of the fact that it doesn't obliges administration and controlling on more level execution at client level. Nonetheless, effective resource provisioning is a key test for cloud computing and determining such sort of issue can diminish under or over utilization of resources, increment client fulfilment by serving more clients during peak hours, decrease execution cost for suppliers and service cost for clients. Existing works on cloud computing focuses estimation the accurate capacity needs, static or dynamic VM (Virtual Machine) creation and scheduling. In any case significant measure of time is obliged to make and devastate VMs which could be utilized to serve more client requests. In this proposal, a QoS (Quality of Service) aware VM provisioning component is developed that guarantees efficient use of the system resources. The VM for comparable kind of request has been reused so that the VM creation time could be minimized and used to serve more clients re-quests. In the proposed model, QoS is ensured by serving all the tasks within the requirements described in SLA. The advances of virtualization technology in Data Centers lead to the development of features such as High Availability and Fault Tolerance. This thesis also presents the enablement solution that covers VM migration in private clouds.

Keywords: Cloud Computing, Service Level Agreement (SLA), Quality of Service (QoS), Virtual Machine (VM), Fault Tolerance.

1. INTRODUCTION

Over the last few years, cloud computing has been paid wide attention by academic organizations, government as well as small, medium and large scale industries. The providers of cloud computing technology over different kinds of ser-vices to users which include programs, storage, application-development platforms over the Internet, hardware resources for deploying user friendly platform etc. Users can access cloud computing services using a variety of devices including PCs, Smart Phones, laptops, PDAs etc.

The services provided by cloud platform are still ongoing and reliability is one of the major goal than services in grid computing and needs more scalability than services provisioned in large commodity clusters. Cloud computing still encounters a number of challenges including clients' model of virtual machine provisioning which will ensure QoS (Quality of Service). Achieving QoS includes a number of parameters and properties to be fulfilled include user experience as well as degree of satisfaction, response time, trust, privacy concern etc.

To enhance user satisfaction and to justify the investment in cloud based deployments, QoS parameters are highly desirable. Some existing research works on QoS[1]-[2] and methods have tried to provide assurance in meeting the SLA (Service Level Agreement). Some other works including [2] tried to control VM provisioning in proactive or reactive manner. However, the target of fulfilling SLA is a great challenge because of the uncertain and dynamic characteristics of network and IT resources in the distributed cloud platform.

2. ACHIEVING RELIABILITY BY FAULT HANDLING

As IT frameworks have gotten to be progressively basic to the smooth operation of an organization, and apparently the economy overall, the significance of guaranteeing the continued operation of those frameworks, and their rapid recovery, has increased. For example, of organizations that had a significant misfortune of business information, 43 percent never revive and 29 percent close inside two years. As a result, planning for continuation or recovery of frameworks needs to be considered exceptionally important. This includes a critical financing of time and cash with the point of guaranteeing insignificant misfortunes in the occasion of a problematic occasion.

2.1 Type of Faults

Site Destruction/Disruption
- Fires, Power and Communications Failure

Mechanical Equipment Failure
- Processor, Disk and Network Hardware

Software Failure
- Operating System and Application

2.2 MTBF, MTTR

Mean Time between Failure (MTBF) is the average amount of time that a device or product functions before failing. This unit of estimation incorporates just operational time between failures and does exclude repair times, expecting the thing is repaired and starts working once more.

MTBF figures are frequently used to extend how likely a single unit is to fail within a certain period of time.
Three quantities are required:
n = Number of observations. u_i = i^{th} Uptime
d_i = i^{th} Downtime following the i^{th} Uptime
So Mean Time between Failures = Sum (d_i -u_i)/ n,
for all i = 1 through n observations.

More simply, it is the total working time divided by the number of failures.
Mean Time to Repair (MTTR) is the standard measure for how quickly the IT organization is able to respond to and resolve issues or problems experienced by its customers. The MTTR can be calculated by dividing the total time required for maintenance by the total number of repairs within a specific timeframe. It is also an important consideration when negotiating service-level agreement (SLA).
Mean Time to Repair = Sum (d_i) / (n)
More simply, it is the average time that it takes to repair something after a failure.

3. RELATED WORK

In [1] author proposed an algorithm which considered Pre-emptable task execution and multiple SLA parameters such as memory, network bandwidth, and required CPU time. It portrays the SLA based resource provisioning and on-line adaptive scheduling for Preemptable task execution. It likewise gives dynamic resource provisioning assignment of additional resources, creation and movement of VMs, alertly reacts to fluctuating workload. These two systems which are joined together in proposed dynamically for effective usage of cloud resources to meet the SLA objective.

In [3] author has created an adaptive QoS (Quality of Ser-vice) aware VM provisioning mechanism that guarantees efficient usage of the system resources. The VM for comparable type of requests has been reused so that the VM creation time could be minimized and used to serve more clients requests. In the proposed model, QoS is guaranteed by serving the entire task inside the necessities depicted in SLA. The model finds out that target QoS has been met by controlling the permission of the demands so that the system does not get over-burden. The model likewise serves to guarantee budget minimization by the enhanced provisioning of IT resources.

In [4] this paper presents challenges, vision, and architectural parameters of SLA-oriented resource management. The proposed architecture supports integration of market based provisioning and virtualisation technologies for efficient provisioning of resources to applications. By utilizing SLAs to portray service quality parameters that are needed by the cloud clients, the Cloud suppliers know how clients fulfilled by their services and consequently can give criticism components to encourage and discourage service demand submissions.

In [5] an autonomic resource director is introduced to control the virtualized environment which decouples the provisioning of resource from the dynamic placement of virtual machines. This manager expects to enhance a worldwide utility capacity which incorporates both the level of SLA satisfaction and the working expenses. It gives automatic dynamic provisioning and placement of Virtual machines considering both provision level SLAs and resource administration costs with high-level monitoring for the manager to determine exchange offs between the two.

In [6] author has proposed an autonomic resource manager to control the virtualized environment which decouples the provisioning of resources from the dynamic placement of virtual machines. This manager aims to optimize a global utility function which integrates both the degree of SLA fulfilment and the operating costs. It relies on two-level architecture with a clear separation between application specific functions and a generic global decision level. In Local Decision Module application constraints are expressed as follows:-

$n_{ik} \leq n_{ik}^{max} \quad 1 \leq i \leq m$ and $1 \leq k \leq c$ (1)

$\sum_{K=1}^{C} n_{ik} \leq T_i^{max} \quad 1 \leq i \leq m$ (2)

Global Decision Module:

U_{global}=*maximize*

$\sum_{K=1}^{C} \alpha i \times \mu i - \epsilon.\mathrm{cost}(Ni) \quad 0 \leq \alpha i \leq 1$ (3)

4. PROPOSED METHOD

In this section, a brief description of the working environment, the assumptions and meaning of the notations used for describing different parameter are provided. The system consists of a set of data centres named SDS. Each of the data centres contains n physical servers. The set of physical server is given by SSS. It is assumed that each of these servers has equal capacity of computing resources (e.g., servers, net-works, storage, applications, and services etc.). The set of application instances is given by SAS and the set of virtual machine is given by SVS. The number of virtual machines required for serving an application depends on the application type and workload variance of the application with time. T_{reqneg} is the requested negotiated time within which the service needs to be provided. $T_{service}$ is the maximum af-fordable time provided by the service provider. $T_{require}$ is the estimated require time to complete the task. $T_{totalservice}$ is the total time requires completing all the tasks in the service and the tasks in the queue waiting for execution. $T_{reserved}$ is the time to reserve VM as per the requirement of the user.

5. RESULTS AND OBSERVATIONS

Cloud computing system can be divided into two sections, one is Frontend and other is Worker-node. The frontend gets the request from the client and accordingly fulfils the requirement by getting the resources form the worker nodes.

Algorithm 1: VM Provisioning

```
INPUT: User Service Request
OUTPUT: VM Provisioning
1: Sort
Student VM List By Key value (student VMist; key Value)
2: Sort
Research VMList By Keyvalue(researcherVMList; keyValue)
3: while VM List is not empty do
4:    for all each value of VM List do
  5:       HostInfo ← Get host information
   6:             if more resources available then
   7:    Calculate priority based on User service credit
   8:            else
   9:    Wait for VM of similar type to complete current job
  10:            end if
  11:    Set vmTemplate values based on the user input.
  12:    Set imageId   Get imageId from the image list
  13:  // Allocate requested VM Template to client
 14:      Rc ←virtualMachine:allocate(oneclient; vmTemplate)
 15:          if Rc has Error then
 16:      Throw exception with rc:getErrorMessage()
 17:         else
 18:      Print allocated VM's ID
 19:        end if
 20:    //Create a representation for the new VM using the returned VM-ID
 21:    Vm←new VirtualMachine(newVMID, oneClient)
 22:    Rc← deploy VM
 23:          if Rc has Error then
 24:      Throw exception with rc:getErrorMessage()
 25:          Else
 26:      Set vmAllocationList to new VM Details
 27:      StartTime←CURRENTTIME
 28:         End if
 29:   End for
 30:   End while
```

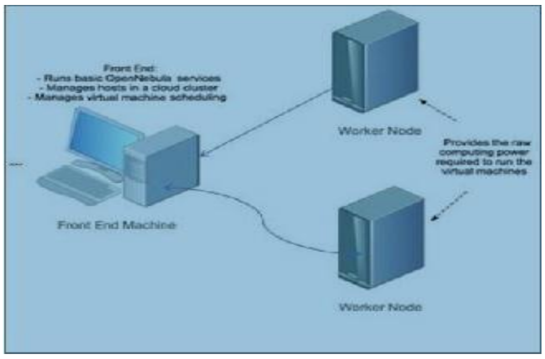
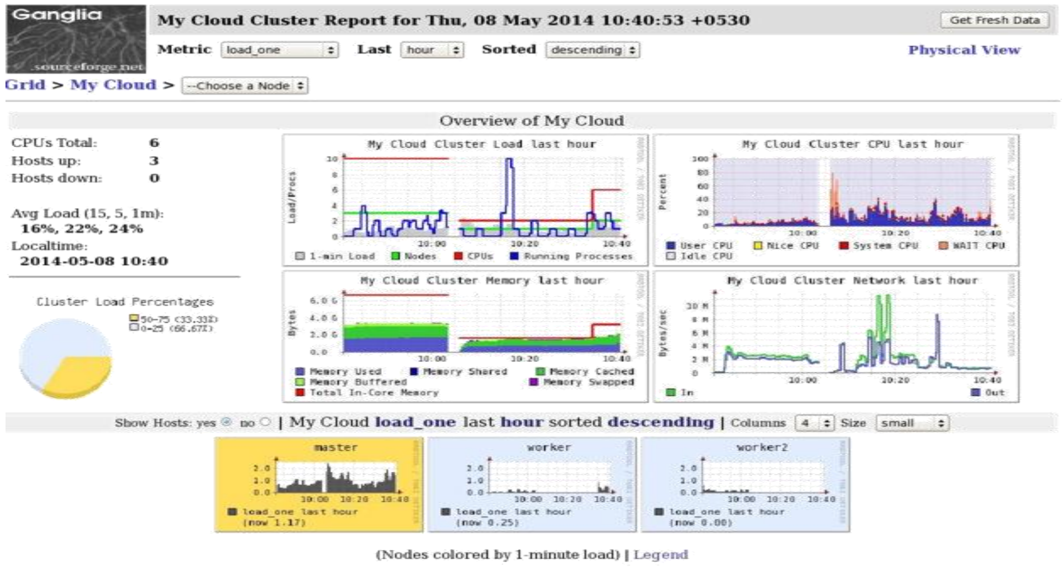

Front End:
- Runs basic OpenNebula services
- Manages hosts in a cloud cluster
- Manages virtual machine scheduling
Worker Node
Provides the raw computing power required to run the virtual machines
Front End Machine
Worker Node

Figure 1: Basic Cloud Scenario

Algorithm 2: Usage Time Measurement

```
1.      TotalTime ←0
2.        Temp← 0
3.        Monitor
4.      while VM state is running do
5.          Monitor resources on host
6.          while VM state is not running do
7.              EndTime←CURRENTTIME
8.        Temp = EndTime – StartTime
9.        TotalTime = TotalTime + Temp
10.             if VM state is ShutDown then
11.               Remove the current VM from vmAllocationList
12.               Return Total Used Time
13.                 Break;
14.             else
15.                 Goto Monitor
16.             end if
17.           end while
18.       end while
```


My Cloud Cluster Report for Thu, 08 May 2014 10:40:53 +0530

Metric load_one Last hour Sorted descending

Grid > My Cloud > --Choose a Node

Overview of My Cloud

CPUs Total: 6
Hosts up: 3
Hosts down: 0

Avg Load (15, 5, 1m):
16%, 22%, 24%
Localtime:
2014-05-08 10:40

Show Hosts: yes no | My Cloud load_one last hour sorted descending | Columns 4 Size small

(Nodes colored by 1-minute load) | Legend

Figure 2: Overview Of Cloud Environment

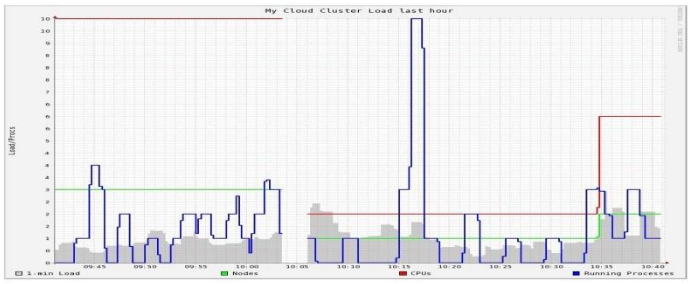


Figure 3: Cloud Load by VMs using existing algorithm of Opennebula

6. CONCLUSION

There is no doubt that the cloud computing is the development trend in the future. Cloud computing brings us the approximately infinite computing capability, good scalability, service on-demand and so on. This work presents an approach for VM and QoS provisioning system for Cloud Environment. It defines the problem of making VM repeatedly and stated a QoS model to recover this problem. The goal of the model is to meet QoS parameters (relibility, scalability, response time) by optimizing rejection time in clouds. Client use cloud instance for many purpose for as parallel rendering, web server or for file sharing etc , Any of this instance have specific softwares for fulfill this requirement and services. When fault occurs, client loss their data but with proper Fault Tolerance plan cloud vender can able to backup all instance image at another site(backup site) and so fault harms to only processing center. Future work can build this Fault tolerance system more optimized and powerfull. Decreasing the time taken by algorithm for migration. Convert this whole architecture in fiber channel

7. REFERENCES

1. Rajnikant B. Wagh Chandrashekhar S. Pawar. Priority based dynamic resource allocation in cloud computing. Cloud and Services Computing (ISCOS), 2012.
2. R. Buyya R. N. Calheiros, R. Ranjan. Virtual machine provisioning based on analytical performance and qos in cloud computing environments. Parallel Processing (ICPP), 2011 International Conference, 2011.
3. Tamal Adhikary Amit Kumar Das. An intelligent approach for virtual machine and qos provisioning in cloud computing. Information Networking (ICOIN), 2013.
4. Rodrigo N. Calheiros Rajkumar Buyya, Saurabh Kumar Garg. Sla-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. Cloud and Service Computing (CSC), International Conference, 2011.
5. Saurabh Kumar Garg Rajkumar Buyya. Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2011.