Enhanced Load Balancing Mechanism with Service Level Agreement using Refuge Console in Private Cloud Environment

[1]M.Murali, [1]M.Saravana karthikeyan, and [2]Dr.S.Sujatha, *Member,* IAENG

Abstract—**Cloud Computing affords an immense collection of computational resources as a service. It has been a possible solution for providing elastic, on insist computing infrastructure for a large number of applications. The applications are created in the private cloud and it can be executed with respect to the refuge, cost and execution time which will be maintained by Service Level Agreements (SLA) with respect to the customers and the service providers. Some of the failures like system crash, hardware and software failures might be the barrier for the success of cloud concepts. In this paper we proposed the concept of private cloud managed services to overcome the security issues. In case of hurtle, change in components, service manage themselves. Based on the concept of effective load management we derived Refuge controlled Private Cloud Service execution to obtain efficient scheduling and load balancing mechanism in cloud environment.**

Index Terms— **Private Cloud, Service Level Agreement, Refuge control.**

I.INTRODUCTION

Cloud computing offers computational energy on Web 2.0 style applications. Private Cloud provides us the Immediacy, Compatible technology, Rapid deployment, Optimize resources, Elasticity, Bursting, Security, and Trust. Various works has been done in the Cloud Computing area. Deploying the services we created in the cloud environment can be used by the customers who are indeed of the respective service after the completion of the concession (negotiation) phase. The infrastructure of cloud computing allows enterprises to attain more proficient use of their applications.

Attention has been paid to the appropriate management of a single Cloud service. Actual customer who needs for the respective service will establish Service Level Agreements (SLAs) with Service providers of the Private Cloud Environment which guarantee the ideal service. In the SLA agreement they speak about cost, quality of service, job duration.

In this paper we present novel concepts for Refuge controlled Private Cloud Service (RCPCS), to solve the issues like how the services are agreed to the customers, what are all the concession protocols used to acquire the service. Based on Concession bootstrapping and service mediation (SM) even diverse service providers and users can map between conflicting SLA templates and negotiate with partners even in case of conflicting SLA templates[6]. e.g., SLA agreement using the principles f Service Mediation(SM) [8].

This paper contributes i) Discussion on the private cloud management services ii) Refuge console role in Private Cloud iii) Management console role in Private Cloud.

II.RELATED WORK

At Present, the works on Private Cloud services can be classified in following categories: (i) narrative explanations for Cloud computing approach (ii) SLAs and concession in related areas as on Grid Computing, Utility Computing (iii) Restrain of Cloud computing.

Many of the IT companies are switched in to the cloud computing area there by their application services are stored secured third party in the cloud servers. If they want to store it locally means the cost

Manuscript received January 20, 2010.This work was supported in part by students and faculties of Pervasive Computing Technologies, Anna University Tiruchirappalli, Tiruchirappalli, India.

M.Murali is doing M.E Pervasive Computing Technologies in Anna University Tiruchirappalli, Tiruchirappalli, India (email: murali49@gmail.com)

M.Saravana karthikeyan is doing M.E Pervasive Computing Technologies in Anna University Tiruchirappalli, Tiruchirappalli, India (email: msk.pervasive@gmail.com)

Dr.S.Sujatha is an associate professor in Anna University Tiruchirappalli, Tiruchirappalli, India (ssujatha71@yahoo.co.in)

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

of buying the hardware and maintaining the service will increase the cost by double. So they need the hands of cloud computing service providers in order to reduce the cost and maintenance time.

III. ARCHITECTURE OF PRIVATE CLOUD MANAGED SERVERS WITH REFUGE CONSOLE

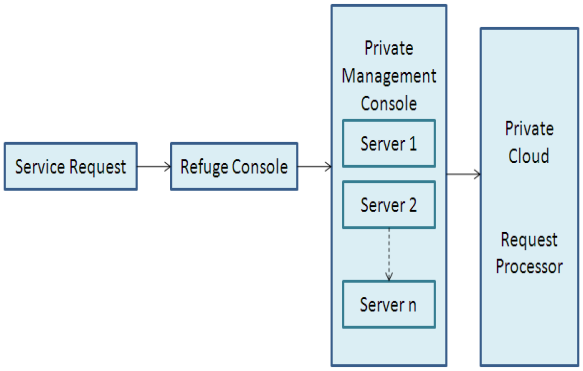

Figure 1 Private Cloud Managed Servers with Refuge Console

Initially the customer who requests for a service in the private cloud environment has to follow the steps to get the service in the private cloud [1]. First the request is pass through the refuge console. The component of the refuge console is given above in Figure 1.

A. Refuge Console: The request for service from the customer is taken and it has to pass through the modules in the refuge console of the private cloud environment which checks the user is an authenticated user or not.

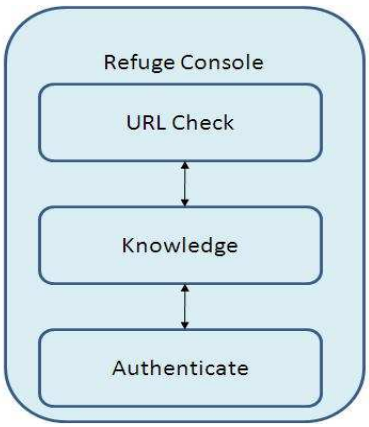

Figure 2 Refuge Console

Checks the URL	Where the request from
Validation	User credential

As mentioned in the Figure 2 first the URL is checked whether the customer has enough user credentials or not. Based on the knowledge the URL is analyzed and it will be authenticated and preceded for further process. If new URL is found, it has to obtain the authentication from refuge console by getting credential accounts. Finally the service requested will pass through management console for further processing.

B. Private Management Console: Management console will guide the authenticated service request to proceed further. Here the available services are listed by the Service Lister and it will be viewed by the customer. Service Lister will notify the conditions to access the services in the request responder console.

Load Balancer: It balances the request load in the management console and gives the weight age to each individual server. The Private Management Console is shown in the Figure 3.

Scheduler: In the private management console the scheduler who schedules the job to all the servers in the request responder thereby the job is given to the respective servers [2], [3].

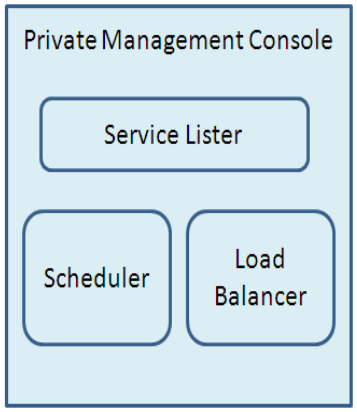

Figure 3 Private Management Console

Services Lister: Services are deployed in the Private Cloud and it is ready to serve based on the customer will. These services are published and it is ready to grab by the customer.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Negotiation Process: Based on the previous usage of the services the customer will view the blog. If he wants to proceed with the same previous service means there will be no need of detailed explanation about the factors like cost of usage, quality of service and duration of service. If he is new means he will be given the detailed explanation about the service usage conditions in the private cloud environment. Like cost, quality of service, service usage duration and everything will be detailed. Based on the constraint given by the service provider particular services are only selected [6].

Complete Negotiation Phase: In this phase the providing and consuming bodies will get all the concession protocols supported with them regarding the service process and completes the negotiation phase. Briefly, job demands are noted like uploading & downloading the information and so on. Finally they will be getting the necessary protocols supported and complete the execution of the operation.

Figure 4 represents the sample Meta Negotiation document [6] and the conditions have to be satisfied before the <pre-requisite> element. The security element is checked before the start of the process, Cloud Security Infrastructure (CSI). There by an electronic contract is signed between two parties till the completion of the operation. If the required platform matches the provider's then the job starts [7], [8].

```
1.  meta-negotiation>
2.  ……
3.  <pre-requisite>
4.  <security>
5.  <authentication value="CSI"
6.   location="uri"/>
7.  </security>
8.  <negotiation-term name="start"/>
9.  <negotiation-term name="end"/>
10. <negotiation-term name="cost"/>
11. </negotiation-terms>
12. </pre-requisite>
13. <negotiation>
14. <document name="WSLA"value="uri"…. >
15. <protocol name="RelevantOffers"
16. schema="uri" version="1.0"location="uri"/>
17. </negotiation>
18. <agreement>
19. <confirmation name="arbitrationService"
20. value="uri"/>
21. </agreement>
22. </meta-negotiation>
```

Figure 4 Example document for meta-negotiation

Request Processor: After the concession process is completed, the request will be getting necessary service from the list of servers available in the request processor. The total work is calculated by the work console [4], [5].

Work console is calculated by means of necessary factors like Job Duration (JD), cost (C) and quality of the service (Q)

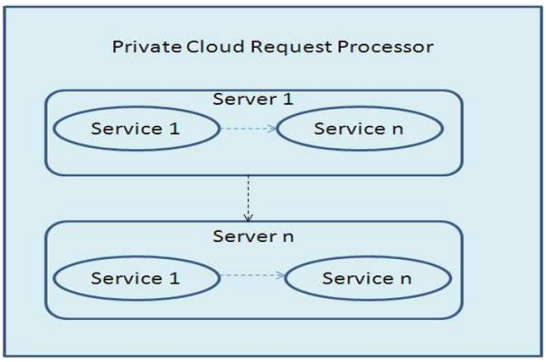

Figure 5 Private Cloud Request Processor

Work Console	JD+C+Q

IV. IMPLEMENTATION AND RESULT

We designed and implemented the Refuge Controller, in which the URL is checked first and the validation completes with the help of knowledge based system. So, the direct hit to the Private Cloud servers will be avoided. There by the over head caused by the unwanted load is reduced for the effective computation of Private Cloud Servers. In the below chart it is clearly explained, how the actual cloud computing environment behaves to the requests with and without Refuge Controller. Experimental result shows that the refuge controller made the Private cloud environment more efficient with effective load balancing.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

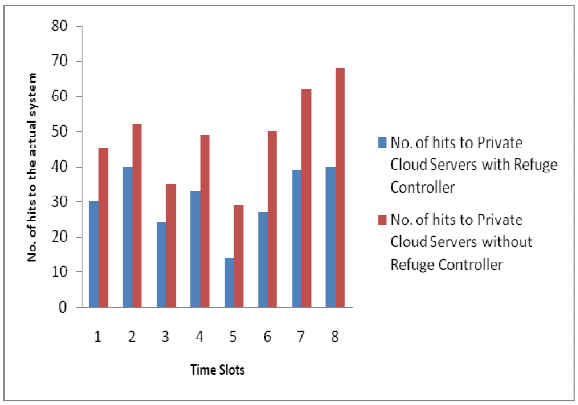

Figure 5 Hits to the Private Cloud Servers

V. CONCLUSION AND FUTURE WORK

In this paper we have implemented the concept of Private Cloud Managed Servers with refuge controller. This gives how the request is taken from the customer with the help of refuge console and then preceded to the management console further to the request responder to complete the service request from the customer.

In future we planned to study deeply the individual components of the private cloud request processor in order to enhance its capabilities with high degree of scalable performance.

REFERENCES

[1]. Sukhyun Song, Kyung Dong Ryu, Dilma Da Silva," Blue Eyes: Scalable and Reliable System Management for Cloud Computing" IEEE International Symposium on Parallel & Distributed Processing, ISBN: 978-1-4244-3751-1, May 2009.

[2]. Nicolas G. Grounds, John K. Antonio and Jeff Muehring, "Cost-Minimizing Scheduling of Workflows on a Cloud of Memory Managed Multicore Machines", Springer Berlin / Heidelberg pages 435-450, 2009.

[3]. Liu, Ke ," Scheduling algorithms for instance-intensive cloud workflows" Swinburne University of Technology. Faculty of Information and Communication Technologies. Centre for Complex Software Systems and Services,2009.

[4]. Christian Vecchiola, Suraj Pandey, and Rajkumar Buyya," High-Performance Cloud Computing: A View of Scientific Applications", Cloud computing and Distributed Systems (CLOUDS) Laboratory, The University of Melbourne, Australia, 2009.

[5]. D. A. Menasce. Scaling Web Sites ThroughCaching. *IEEE Internet Computing*, July/August 2003. A.Ninan, P. Kulkarni, P. Shenoy, K. Ramamritham, and R. Tewari. Scalable Consistency Maintenance in Content Distribution Networks Using Cooperative Leases. *IEEETKDE*, July 2003.

[6]. Ivona Brandic. Distributed System group, Institute of Information Systems," Towards Self-manageable Cloud Services", 2009 33rd Annual IEEE International Computer Software and Applications Conference.

[7]. D. Nurmi, R. Wolski, Ch. Grzegorczyk, G. Obertelli, S.Soman, L. Youseff, D. Zagorodnov. The Eucalyptus Open-source Cloud- computing System. Proceedings of Cloud Computing and Its Applications 2008, Chicago, Illinois, October 2008.

[8]. Web Service Level Agreement (WSLA), http://www.research.ibm.com/WSLASpecV1-20030128.pdf

AUTHORS

M.Murali is doing M.E Pervasive Computing Technologies in Anna University Tiruchirappalli, Tiruchirappalli, India. He received B.E Computer Science and Engineering from Anna University, Chennai, India in the year 2007. He is a member of IAENG and IACSIT.

M.Saravana karthikeyan is doing M.E Pervasive Computing Technologies in Anna University Tiruchirappalli, Tiruchirappalli, India. He received B.Tech Information Technology from Anna University, Chennai, India in the year 2005. From 2006 to 2008 he worked as a System Engineer for Anna University, Chennai. He is a member of IAENG

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)