International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 04, April 2017
www.ijates.com
ijates
ISSN 2348 - 7550

BUILDING VIRTUAL MACHINE INSTANCE, COMPATIBLE WITH USER'S WEB APPLICATION IN OPEN STACK CLOUD PROVIDER

Mrs.Niranjana A[1], Ms.A.Geetha[2]

Assistant Professor, Department of Computer Science and Engineering

Easwari Engineering College, Chennai

ABSTRACT

One of the key challenges in building a platform for deploying applications is to automatically compose, configure, and deploy the necessary application that consists of a number of different components. Setting up complex combination of appliances is costly and error prone even in traditional hosting environments. Virtual appliances provide an elegant solution for this problem. They are built and configured with a necessary operating system and software packages to meet software requirements of a user. In Existing System, non-expert users are not aware of composition of cloud services, whether applications are compatible or not. They are not aware of web application dependencies and configuration details. Real time process of virtual instances is not supported. Individual appliances cannot be simply put together as they may not be compatible with the hosting environment. None of the providers provide any ranking system to choose the best instance type and software solution for the deployment. OpenStack Cloud provider which is the best to build a Virtual Instance is used in this system. This system helps non-expert users with limited or no knowledge on legal and image format compatibility issues to deploy their services faultlessly. Users have the flexibility to build their own instance based on his/her requirements. Even for web applications, user can select their web application dependencies based on his/her preferences.

Key Terms: Virtual Instance, Web Application, OpenStack, Cloud Services, Virtual Appliance

I INTRODUCTION

A single Cloud service (i.e., a software image and a virtual machine), on its own, cannot satisfy all the user requirements, a composition of Cloud services is required. Cloud service composition, which includes several tasks such as discovery, compatibility checking, selection, and deployment, is a complex process and users find it difficult to select the best one among the hundreds, if not thousands, of possible compositions available. Service composition in Cloud raises even new challenges caused by diversity of users with different expertise requiring their applications to be deployed across difference geographical locations with distinct legal constraints. The main difficulty lies in selecting a combination of virtual appliances (software images) and infrastructure services that are compatible and satisfy a user with vague preferences.

One of the key challenges in building a platform for deploying applications is to automatically compose, configure, and deploy the necessary application that consists of a number of different components. The deployment requirements of a web application service provider will include security devices (e.g. firewall), load balancers, web servers, application servers, database servers, and storage devices. Setting up such a complex combination of appliances is costly and error prone even in traditional hosting environments. Virtual appliances provide an elegant solution for this problem. They are built and configured with a necessary operating system and software packages to meet software requirements of a user. Non-expert users are not aware of composition of cloud services, which are compatible or not. They are not aware of web applications dependencies and Configuration details. Their deployed applications can be misconfigured. Real time process of virtual instances is not supported. Individual appliances cannot be simply put together as they may not be compatible with the hosting environment. Setting up complex combination of appliances is costly and error prone. None of the providers provide any ranking system to choose the best instance type and software solution for the deployment.

II RELATED WORKS

Amir Vahid Dastjerdi (2014) deduced that a single Cloud service (i.e., a software image and a virtual machine), on its own, cannot satisfy all the user requirements, a composition of Cloud services is required. Cloud service composition, which includes several tasks such as discovery, compatibility checking, selection, and deployment, is a complex process and users find it difficult to select the best one among the hundreds, if not thousands, of possible compositions available. Service composition in Cloud raises even new challenges caused by diversity of users with different expertise requiring their applications to be deployed across difference geographical locations with distinct legal constraints. The main difficulty lies in selecting. a combination of virtual appliances (software images) and infrastructure services that are compatible and satisfy a user with vague preferences.Therefore, they present a framework and algorithms which simplify Cloud service composition for unskilled users. They developed an ontology-based approach to analyze Cloud service compatibility by applying reasoning on the expert knowledge. In addition, to minimize effort of users in expressing their preferences, they apply combination of evolutionary algorithms and fuzzy logic for composition optimization. This lets users express their needs in linguistics terms which brings a great comfort to them compared to systems that force users to assign exact weights for all preferences.

Changhua Sun (2008) focused on simplifying service deployment with virtual appliances. Techniques used by them are the use of virtual appliances as a better service for deployment and comparison of traditional deployment mechanisms and virtual appliances type of deployment. They start with an easy to understand model to describe the complexity of service deployment and introduce the architecture of a virtual appliance. It is observed that the deployment process of using traditional deployment mechanisms, and quantitatively and qualitatively compares operations and parameters of traditional approach with the use of virtual appliances. The results show virtual appliances offer significant advantages for service deployment by making deployment process much simpler and easier, even for the deployment of advanced enterprise services.

Rajkumar Buyya (2010) observed that Virtual Appliances, pre-configured, ready-to-run applications are emerging as a breakthrough technology to solve the complexities of service deployment on Cloud infrastructure. However, an automated approach to deploy required appliances on the most suitable Cloud infrastructure is neglected by previous works which is the focus of this work. In this paper, they proposed an effective architecture using ontology-based discovery to provide QoS aware deployment of appliances on Cloud service providers. In addition, they tested their approach on a case study and the result shows the efficiency and effectiveness of the proposed work.

III PROPOSED SYSTEM

Proposed system uses OpenStack Cloud provider which is the best to build Virtual Instance. OpenStack software controls large pools of compute, storage, and networking resources throughout a datacenter, managed through a dashboard or via the OpenStack API. Cloud services consisting of virtual appliance and units are compatible with each other. The system helps non-expert users with limited or no knowledge on legal and image format compatibility issues to deploy their services faultlessly. User can build their own instance based on his/her requirements. Web applications dependencies are available, which means the user only has to select his web application dependencies based on his/her preferences.

Initially, the user buys images based on personal requirements which includes Operating Systems, RAM, and Database. Following this, Compatibility checking of cloud services is done. If the cloud services are compatible, the user is redirected into GSVC Internet Banking. After successful transaction, the amount will be debited from his/her account. The user checks the newly created instances in launching an Instance module. Information about User owned instances details are stored in database. Following that, instances owned by the user and running instances are displayed. Using jclouds, user can launch and stop the instances. User can check their active instances in web page or in OpenStack Dashboard. The user can finally launch his web application with the help of Putty.

The proposed system consists of the following modules

- Cloud Service Composition
- Build and Configure Instance
- Launch an Instance
- Deploy Application

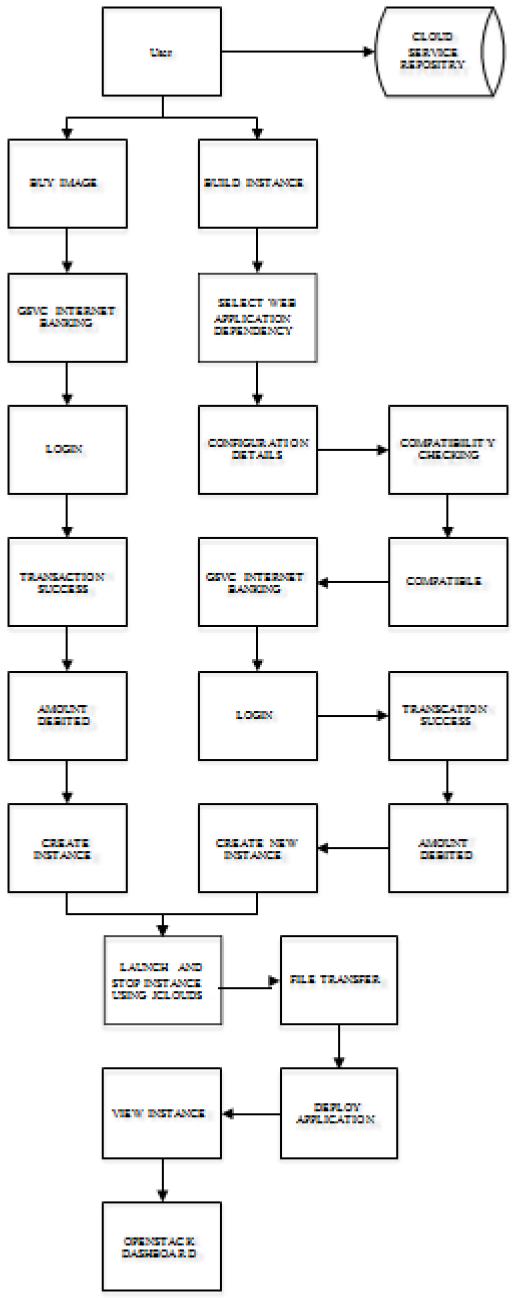


Fig. 1 Overview of System Design

Cloud Service Composition

The user has to register his/her details like Username, Date of Birth, Contact Number, Email etc. on the main page. User ID is automatically generated when the user presses the Click Button. If the user has already registered his details, the user can directly log in. The server in turn stores the user information in its database. User can also view and modify their information under the tab My Profile and updated information is stored in the database.

Advertisement of two default instances of service provider is available in this module.

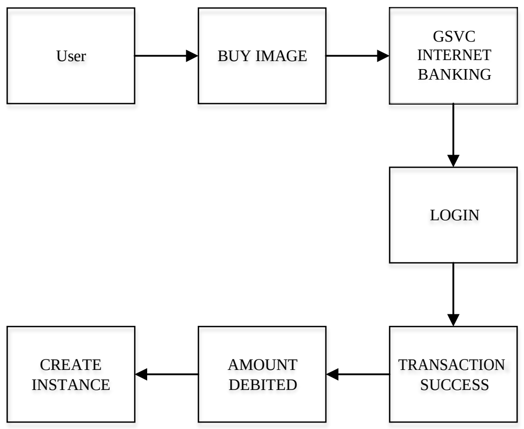


Fig. 2 Cloud Service Composition

Default instances consist of details like OS, RAM, Web Server, Database and Price. User buys instances based on their requirements. At buy instance stage, user is redirected into GSVC Internet Banking Web Page. The user has to enter his Bank account User ID and password in order to log in. The user confirms his purchase by checking his balance and entering his transaction password. After successful transaction, the amount will be debited from his/her account. The user checks their bought instances in launching an Instance module. User owned instances details are stored in database.

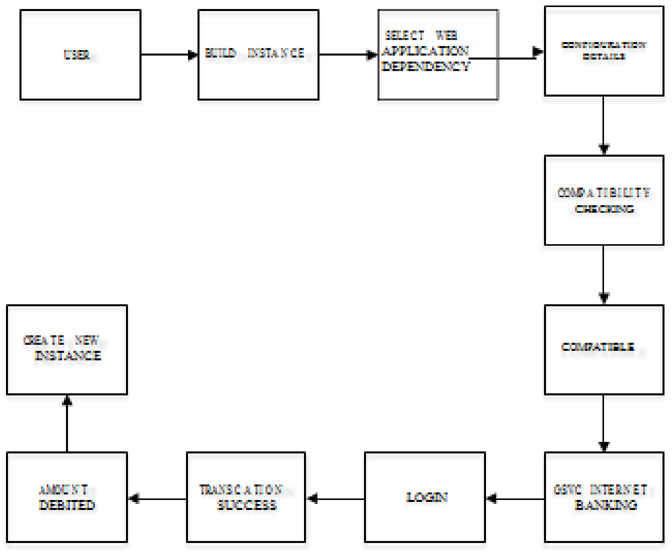
Build and Configure Instance

User can create new instance based on his/her requirements. In this module, user can select their Web application dependency which includes Operating Systems, RAM, and Database, Architecture and Budget. Requested Service will not be provided if there are budget constraints. Therefore, using high-level if-else cases, Compatibility checking of cloud composition services is done successfully. The user is then redirected to the Web Page which gets the type and weight of Web Application to be deployed from the user. Selection of RAM and whether the web application needs a database is also checked. If the cloud services are fully compatible, user is redirected into GSVC Internet Banking.

.The user has to enter his Bank account User ID and password in order to log in. The user confirms his purchase by checking his balance and entering his transaction password after successful transaction, the amount will be debited from his/her account. The user checks their newly created instances in launching an Instance module.

Launch Instances

In this module, instances owned by the user and running instances are displayed. Now user can launch any instances. An Instance is launched using Apache jclouds, (which is an open source multi-cloud toolkit for the Java platform that gives the user the freedom to create applications that are portable across clouds while giving the user full control to use cloud-specific features) based on specified RAM and instance name. Start and end execution time of the instance will be recorded. Instance will be created in OpenStack Dashboard. A separate IP, name and image name will be created for each instance. Now user can view their virtual instance in OpenStack.



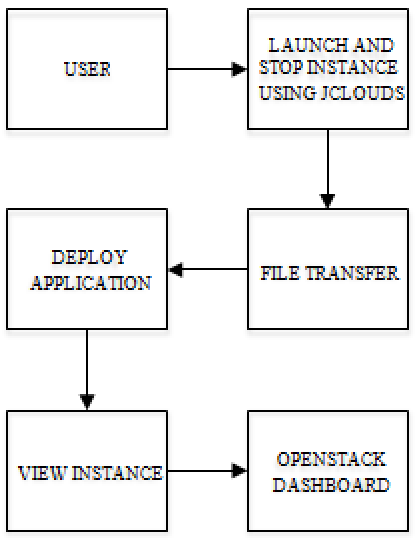


Fig. 3 Build and Configure Instance **Fig. 4 Launch and deploy application**

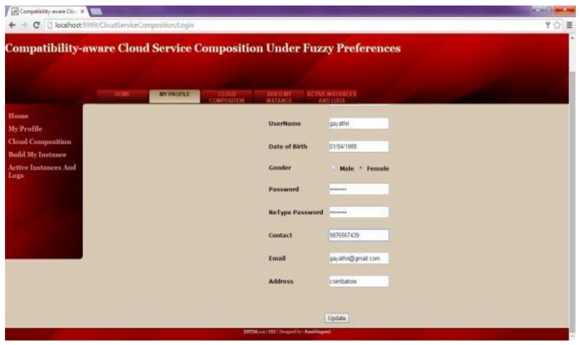
Deploy Application

Using putty, user connects to IP assigned for the virtual instance. Now user connects into Virtual Instance. Giving file transfer command, user transfer their web application to Virtual instance. The syntax for transferring file is < pscp – sftp filename username@instanceipaddress> and for transferring a folder is <pscp –stfp –r folder name username@instanceipaddress> User can also transfer database files to Virtual Instance. Now user can deploy their web applications in Virtual Instance with the help of Apache Tomcat. Real time process is shown in OpenStack Dashboard.

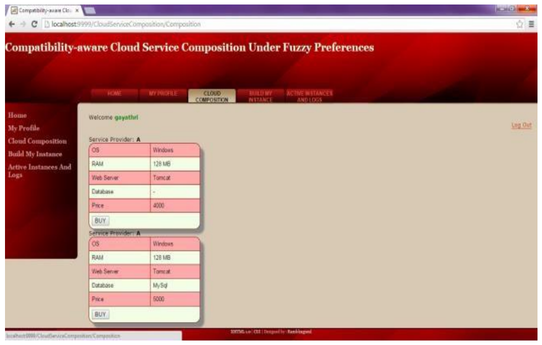
IV. RESULTS AND DISCUSSION

Cloud services consisting of virtual appliance and units are made compatible with each other. This system helps non-expert users with limited or no knowledge on legal and image format compatibility issues to deploy their services faultlessly. User are given freedom to build their own instance based on his/her requirements. Web applications dependencies are available, which means the user only can select his web application dependencies based on his/her preferences.

Initially, the user buys images based on personal requirements which include Operating Systems, RAM, and Database. Following this, Compatibility checking of cloud services is done. If the cloud services are compatible, the user is redirected into GSVC Internet Banking. After successful transaction, the amount will be debited from his/her account. The user checks the newly created instances in launching an Instance module. Information about User owned instances details are stored in database. Following that, instances owned by the user and running instances are displayed. Using jcloud, user can launch and stop the instances. User can check their active instances in web page or in OpenStack Dashboard. The user can finally launch his web application with the help of Putty.


localhost:9999/CloudServiceComposition/Login
Compatibility-aware Cloud Service Composition Under Fuzzy Preferences
HOME
MY PROFILE
CLOUD COMPOSITION
BUILD MY INSTANCE
ACTIVE INSTANCES AND LOGS
Home
My Profile
Cloud Composition
Build My Instance
Active Instances And Logs
Update

UserName	gayathri
Date of Birth	01/04/1988
Gender	Male [✓ Female]
Password	•••••••••
ReType Password	•••••••••
Contact	9876567439
Email	gayathri@gmail.com
Address	coimbatore


localhost:9999/CloudServiceComposition/Composition
Compatibility-aware Cloud Service Composition Under Fuzzy Preferences
HOME
MY PROFILE
CLOUD COMPOSITION
BUILD MY INSTANCE
ACTIVE INSTANCES AND LOGS
Home
My Profile
Cloud Composition
Build My Instance
Active Instances And Logs
Log Out
BUY

Welcome gayathri

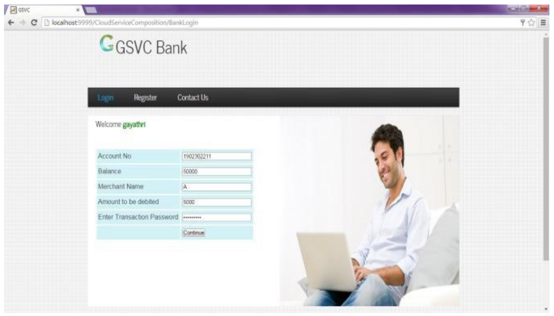
Service Provider: A

OS	Windows
RAM	128 MB
Web Server	Tomcat
Database	-
Price	4000

Service Provider: A

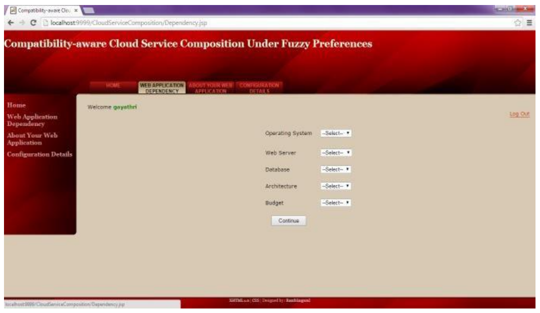
OS	Windows
RAM	128 MB
Web Server	Tomcat
Database	MySql
Price	5000

Fig.5 User Profile **Fig.6 Cloud Composition Details**



Welcome gayathri

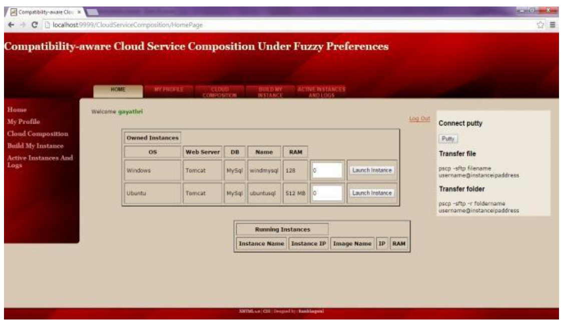
Account No	1902302211
Balance	50000
Merchant Name	A
Amount to be debited	5000
Enter Transaction Password	••••••••••


localhost:9999/CloudServiceComposition/Dependency.jsp
Compatibility-aware Cloud Service Composition Under Fuzzy Preferences
HOME
WEB APPLICATION DEPENDENCY
ABOUT YOUR WEB APPLICATION
CONFIGURATION DETAILS
Home
Web Application Dependency
About Your Web Application
Configuration Details
Log Out
Continue

Welcome gayathri

Operating System	--Select--
Web Server	--Select--
Database	--Select--
Architecture	--Select--
Budget	--Select--

Fig.7 Transaction Page **Fig.8 User Preferred Web Application**


localhost:9999/CloudServiceComposition/HomePage
Compatibility-aware Cloud Service Composition Under Fuzzy Preferences
HOME
MY PROFILE
CLOUD COMPOSITION
BUILD MY INSTANCE
ACTIVE INSTANCES AND LOGS
Home
My Profile
Cloud Composition
Build My Instance
Active Instances And Logs
Log Out
Putty

Welcome gayathri

Owned Instances

OS	Web Server	DB	Name	RAM		
Windows	Tomcat	MySql	windmysql	128	0	Launch Instance
Ubuntu	Tomcat	MySql	ubuntusql	512 MB	0	Launch Instance

Running Instances

Instance Name	Instance IP	Image Name	IP	RAM

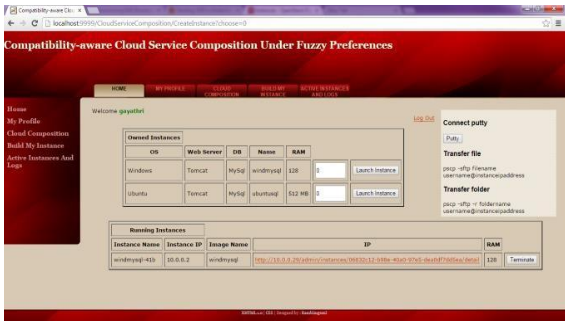
Connect putty

Transfer file

pscp -sftp filename username@instanceipaddress

Transfer folder

pscp -sftp -r foldername username@instanceipaddress


localhost:9999/CloudServiceComposition/CreateInstance?choose=0
Compatibility-aware Cloud Service Composition Under Fuzzy Preferences
HOME
MY PROFILE
CLOUD COMPOSITION
BUILD MY INSTANCE
ACTIVE INSTANCES AND LOGS
Home
My Profile
Cloud Composition
Build My Instance
Active Instances And Logs
Log Out
Putty

Welcome gayathri

Owned Instances

OS	Web Server	DB	Name	RAM		
Windows	Tomcat	MySql	windmysql	128	0	Launch Instance
Ubuntu	Tomcat	MySql	ubuntusql	512 MB	0	Launch Instance

Running Instances

Instance Name	Instance IP	Image Name	IP	RAM	
windmysql-41b	10.0.0.2	windmysql	http://10.0.0.29/admin/instances/06832c12-b98e-40a0-97e5-dea0df7dd5ea/detail	128	Terminate

Connect putty

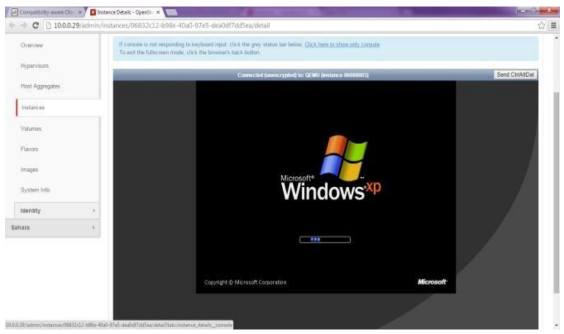
Transfer file

pscp -sftp filename username@instanceipaddress

Transfer folder

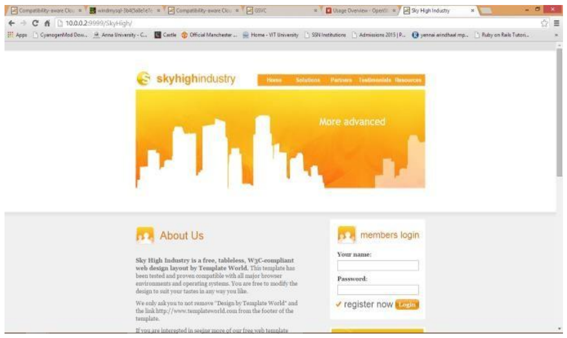
pscp -sftp -r foldername username@instanceipaddress

Fig.9 Display User Instances **Fig.10 Instance Launched**



If console is not responding to keyboard input: click the grey status bar below. Click here to show only console
To exit the fullscreen mode, click the browser's back button.

Connected (unencrypted) to: QEMU (instance 00000003)



Sky High Industry is a free, tableless, W3C-compliant web design layout by Template World. This template has been tested and proven compatible with all major browser environments and operating systems. You are free to modify the design to suit your tastes in any way you like.

We only ask you to not remove "Design by Template World" and the link http://www.templateworld.com from the footer of the template.

If you are interested in seeing more of our free web template

Fig.11 Running the Instance **Fig.12 Deploy Web Application**

[3] Buyya, R. and Dastjerdi, A. (2014) 'Compatibility-Aware Cloud Service Composition under Fuzzy Preferences of Users', in Proceedings of IEEE Transactions on Cloud Computing.

[4] Buyya, R. and Dastjerdi, A. (2012) 'An autonomous reliability-aware negotiation strategy for cloud computing environments,' in Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE.

V. CONCLUSION

In this system Virtual Machine Instance is successfully built in which compatible User Web Application is successfully deployed in Open Stack Cloud Provider. A platform or virtual OS is provided for the user which enables the user to launch a personal webpage with the help of Virtual Box, Apache Tomcat, Putty and MySQL.

Thus the enhancement made solves the issues in the existing system by running complex combination of virtual appliances using OpenStack and thoroughly checking the compatibility of the cloud service composition and deploying applications successfully.

Future enhancement of this system includes, supporting contemporary Operating Systems for the user to work on. Further work can be done to support deployment of multiple web applications in a single instance.

REFERENCES

[1] Bartalos, P. and Bieliková, M. (2012) 'Automatic dynamic web service composition: A survey and problem formalization', Computing and Informatics, Vol.30, No.4, pp.793–827.

[2] Benn, N. Domingue, J. and Lambert, D. (2010) 'Integrating heterogeneous web service styles with flexible semantic web services groundings', in Proceedings of the 1st International Future Enterprise Systems Workshop.

[5] Buyya, R. Dastjerdi, A. and Tabatabaei, S. (2010) 'An effective architecture for automated appliance management system applying ontology-based cloud discovery,' in Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing.

[6] Chen, H. and Yao, Y. (2009) 'Qos-aware service composition using nsga-ii,' in Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, ser. ICIS '09. New York, NY, USA: ACM.

[7] Chieu, T. Karve, A. Mohindra, A. and Segal, A. (2010) 'Solution based deployment of complex application services on a cloud,' in Proceedings of the 2010 IEEE International Conference on Service Operations and Logistics and Informatics.

[8] Durillo, J. and Nebro, A. (2011) 'jmetal: A java framework for multi-objective optimization,' Advances in Engineering Software, Vol.42, No.10, pp.760–771.

[9] Lecue, F. and Mehandjiev, N. (2009) 'Towards Scalability of Quality Driven Semantic Web Service Composition', Proc. IEEE International Conference on Web Services, pp. 469-476.

[10] Sun, C. Wang, Q. and Willenborg, R. (2008) 'Simplifying service deployment with virtual appliances,' in Proceedings of the IEEE International Conference on Services Computing (SCC), IEEE International Conference on Services Computing (SCC).