Embedded Systems Programming in a private cloud

A prototype for "Embedded Cloud Computing"

By

Achin Mishra

Roll Number : 108CS076

B.Tech VIII Semester 2012

May 14, 2012

National Institute of Technology Rourkela,Odisha - 769008

National Institute of Technology Rourkela, Odisha - 769008

Certificate

This is to certify that the work in the thesis entitled Embedded Systems Programming in a private cloud-A prototype for Embedded Cloud Computing submitted by Achin Mishra is a record of an authentic work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at National Institute of Technology, Rourkela.

Place : NIT Rourkela
Date :

Dr. Ashok Kumar Turuk
Associate Professor
Head of the Department
Computer Science and Engineering
NIT Rourkela

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr.Ashok Kumar Turuk,Department of Computer Science and Engineering for his able guidance,wise suggestions and the never ending support and faith,without which my work would not have been possible.

I would also like to extend my thanks to Defence Institute of Advanced Technology (DIAT), Pune for holding a seminar on "Challenges in Cloud computing" in Pune. All the learnings are improvised in the project with their heartly support.

I would also like to express my gratitude towards my father. Along with the supervisor, his didactic words were inspiring that helped me to perform impeccably during the project.

Achin Mishra

(108CS076)

Abstract

Cloud computing is an Internet based computing where resources are shared and maintained at datacenters which could be geographically located across globe. One can rent a virtual server, required memory, OS, applications or load his own softwares on it and access it from any part of the world as per his uses. Data can be stored and are secured in these datacenters which can only be accessed by authorized users. It has changed the total worlds scenario from owning private computers and servers in a private network to on-demand access network of shared computing resources with minimum management burden to prospective users, thus the cloud is an amalgamation of some of the best computing technologies and processes for reliable, resilient and secure applications.

Embedded Cloud computing is the new dimension in cloud computing arising from the merging of the Embedded computing with it. Most of the small embedded systems can be linked (wired/wireless) which can communicate and perform the job. But if those systems might have an upper hand on the cloud, then most of the processing could be shifted to the cloud server and various data centers and henceforth reducing the load of devices. Moreover, devices can be programmed by sitting from any part of the world, without much hustle. Myriad industries have numerous control systems which can be controlled from a central checkpoint.

This thesis exemplifies an attempt to produce a working environment for the same. EDaS (Embedded Device as a Service) is the proposed model of the project which lies at the boundary of IaaS and PaaS.

Contents

1 Introduction **5**

1.1 Cloud Computing 5

1.2 Embedded Computing 5

2 Literature Review **7**

2.1 Cloud Computing Architectures 7

2.2 Current cloud services 9

2.2.1 IaaS 9

2.2.2 PaaS 10

2.2.3 SaaS 11

2.3 Embedded System Design 12

2.4 MBed Prototyping Board 13

3 Existing Cloud based Programming Model **14**

4 Construction of a private cloud **15**

4.1 Considerations for a private cloud 15

4.2 Building of a private cloud 18

4.2.1 Open Source tools 18

4.2.2 Architecture 19

4.2.3 Steps Involved 20

5 Proposed Model **21**

5.1 Creation of Machine Image 21

5.2 Proposed Cloud Architecture 23

6 Conclusion and Future Scope **24**

List of Figures

1 General layout of cloud services 12

2 Existing cloud based model for ES Programming 14

3 Implemented server architecture 19

4 Showing the registration of the created kernel image 21

5 Creating the final registered image 22

6 Proposed architecture for cloud services 23

1 Introduction

1.1 Cloud Computing

Computing has seen changes over decades. Cloud computing wields applications through the internet, which can be reached using a Web browser, while the business pertaining softwares and data are stored on servers at a totally remote location. From mainframe computing to client-server model, it has cleverly picked up similarities from autonomic computing, grid computing, utility computing and many more which have been the major contributors of this master mind. All are tested on private networks and very few on internets, to a small scale, but cloud computing has based itself primarily on Internet. In fact, the cloud is a metaphor for Internet. It has been a revolution in IT, with clear intentions of merging information technology and communication. Cloud computing can be used for the applications on the Internet that store and secure data along with delivering a service, anything including email, automation and tax works. [1]. Sun has coined the phrase The Network is The Computer and they have claimed it as the next generations network computing. But few believe that it has just renamed the common technologies and techniques already existed in the IT sector [2].

1.2 Embedded Computing

An embedded system is a computer system fabricated for specific controlling functions installed in a larger system with real-time computing constraints [4]

[6]. It is embedded as part of a device including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible. At present, embedded systems control many devices in common use today [7].

Embedded Systems and Cloud computing are undoubtedly the two pervasive spheres of computer science across globe. Cloud has shown an exponential growth in industries. Growth of datacenters and encouraging cloud applications has instigated the development of clouds. With an eye on this, embedded systems are now started taking services of the cloud.

2 Literature Review

2.1 Cloud Computing Architectures

We can define the very basic albeit significant architecture of cloud i.e. front end and the back end. The front end is the one perceptible to clients and the back end is the cloud itself comprising storage devices, servers etc. The system architecture and the software communicate with each other through loose coupling mechanism such as messaging queue. But this architecture does not define the needs of any enterprise. Any computing involves a front end and a back end. We must have a more specified, detailed and qualified architecture which connects well.

We have three types of services available in this computing: SaaS (Software as a Service), PaaS (Platform as a Service) and IaaS (Infrastructure as a Service). Embedded devices are also linked up to cloud services and this is considered as EDaS (Embedded Device as a Service) from cloud. The architecture involves a proper functioning of various components like computing and managing of resources within these service layers. Infrastructure as a service certainly needs high technical expertise because the whole infrastructure is made available on-demand basis. Then comes platform as a service and followed by software as a service. Embedded Device as a service is still on the run for a consolidated form.

There are various deliberations for cloud computing architectures but before discussing them, I wish to throw some light on the type of services or architectural layers available. Software as a Service (SaaS) tenders an application to user absolutely on On-Demand basis. Only one instance of the software runs on the cloud and the cloud caters to multiple users or clients i.e. users and applications. The widespread example is salesforce.com. Platform as a Service (PaaS) has softwares which can act as a platform for development of higher level services. That platform may encompass middleware, applications, OS and environment to develop those services. Also the platform can be programmed with help of APIs. Software architects or developers can see only the GUI or API to configure the platform with all the necessary latent tools. These platforms are really capable of handling user applications. Examples of PaaS include Google Apps Engine. Infrastructure as a Service (IaaS) includes fundamental necessities like storage, computing resources etc on the network. Routers, switches, servers, memory etc are made available on network to IT architects to own whole infrastructure without actually installing all those physically. Workloads are well distributed among the components to produce the desire upshot. Various examples of IaaS include Joyent, which deals in producing high performance virtualized servers via On-Demand infrastructure.

Apart from these architectural layers, the cloud is also deployed in various models such as Public, Private and Hybrid Cloud. Public cloud is already out on internet and Private cloud is private to a particular organization. Public

cloud is mostly managed by organizations which deal multiple users at a time with the help of virtualization. Data is secure and multiple requests are dealt at a common platform of cloud. It is much larger than Private cloud. Private cloud is mostly for single client with total control on security, data and service. Hybrid cloud is basically the combination of both public and private clouds. Private cloud is enhanced with resources of a Public cloud and helps to maintain a better reliability and broader functioning of the cloud. Though its a complex structure because it should be judged, the distribution of the application to run on which model, whether Public or Private cloud. It is really advantageous when large resources need small computing on Public cloud.

2.2 Current cloud services

2.2.1 IaaS

Now, its evident that we need certain computing resources and management policies for the same before we rent them. Cloud computing derives this feature from well known Utility Computing. In utility computing, computing resources such as storage and services are charged on the amount of time you use. Like the meters in houses to measure the unit of electricity, water, cooking gas etc. This model has advantages like you need not have to own resources and maintain them, instead rent them. There would not be in any trouble for licensing of softwares, hardwares and other network issues. IaaS requires the benefits from this existing architecture. This is the foundation of cloud services. It needs high technical competency and involves IT architects who can very well provide

Infrastructure as a Service to customers.

It is based on On-Demand model or pay-for-use model. It helps an enterprise to have a virtual infrastructure and services. Customers can handle their infrastructural needs sitting in any part of the world with the help of just one console. A group of IT experts is a necessity to provide infrastructure services with resources like server, storage and network. These resources are the upper layer of infrastructure managed by the user.

2.2.2 PaaS

PaaS, as discussed, includes various development environment and developed applications are brought to live with the help of various disposal environments. PAAS supports fabrication of higher level of services with development and disposal environments acting in synergy. Platforms can be environments such as ASP.NET and provide easy way for vivid business applications. Cloud architecture compatible softwares are main aim of developers using both, development and disposal environment. In this model, as there are many different environments, the applications are specific in nature i.e. they run on particular cloud. Hence many public clouds can be seen in market with their own platforms as the service like Google Apps.

Platform provided to the developers is the core area to design softwares. It provides a sort of framework for applications and hence its not unwise to call PaaS a kind of Framework Computing. Applications designated for definite

platforms cannot run on other platforms. There is no need to worry about the underlying architecture, one can create applications with necessary tools and requirements, and put to use in disposal environment. This facilitates a rapid development and disposal of applications. Development environments include IDEs which are configured or integrated to PaaS for designing, building and validating of applications. Netbeans, Microsoft Visual Studio and others as well are connected to components of PaaS and helps in development of applications on the platform.

2.2.3 SaaS

The model provides built applications to users through a web browser or any other interface applications. These are customizable applications and user is not concerned with the underlying platforms or infrastructure. Various business organizations, in order to meet their business needs, run pertaining applications rather than troubling themselves with developing platforms or infrastructure on which those applications run.

It is often seen that applications are mostly of hosting type like ASP dot com etc but the platform for the application is hidden to the user. Multiple instances are created for the application and customizable for every individual. Applications are, hence, lifeline of this model. Be this application interact with basic infrastructure or platforms, or it runs on a simple server, these are not the matters of concern for users. They just want their request to get delivered without any botheration on how resources are computed and how they are

managed. Applications are designed in a way to give performance, security and useful information to users irrespective of the area from where and when they access.

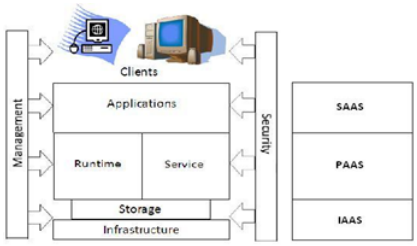

Figure 1: General layout of cloud services

2.3 Embedded System Design

Embedded System Design is unique because it is a hardware-software co-design problem. Both, software and hardware must be designed to together to make sure that the implementation functions properly and is reliable and cost effective [5]. Wayne Wolf describes four major tasks that are

1. Partitioning function to be implemented in interacting and smaller pieces
2. Allocating function is implemented directly in hardware and software running on microprocessor
3. Scheduling function is times at which functions are executed, which is important when single hardware unit is shared

4. Mapping function is functional description for set of components, either on hardware as a logic or in software for a given microprocessor

There are various issues in designing of embedded systems such as hardware software partitioning, fabrication of hardware, software development, code design and simulation etc. Hence an easy prototyping board was chosen for the purpose of interfacing.

2.4 MBed Prototyping Board

The mbed prototyping board, suggested by Joe Bungo [3], is a series of microcontrollers development boards designed for fast, flexible and low-risk and professional rapid prototyping.

The mbed controller has certain packages such as a small 40-pin 0.1" DIP form-factor which convenient for prototyping. It includes a built-in USB programming interface that is as simple as using a USB Flash Drive. Just plug in USB port, copy and paste an ARM program binary, and it starts running [8]. This can re-flash the microcontroller without needing drivers or a programming application. The program binary can be easily generated using the mbed Online Compiler, or alternatively using any other standard tool chain like Keil uVision, Code Red, or GCC. There are also offline compilers such as gcc4mbed, which can be installed and one can export the codes to uVision also. Also, there is support for a virtual serial port using the same USB interface. It enables communication with a PC terminal, Labview, Matlab, and any other programming

language that can communicate with a COM port.

3 Existing Cloud based Programming Model

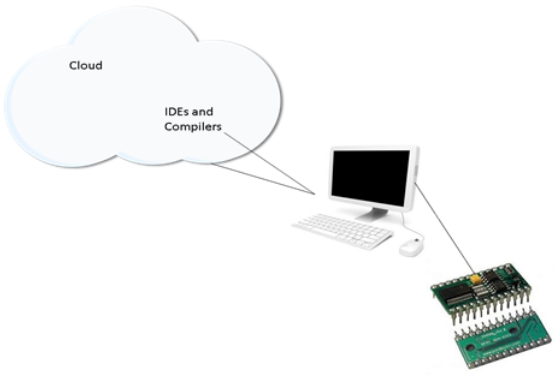

Figure 2: Existing cloud based model for ES Programming

Joe Bungo [3] produces a clear picture on the current cloud based model for embedded systems programming. The host is on the cloud and it utilizes the services of the cloud to program the microcontroller. MBed provides a web interface for programming and downloading of the machine codes from an online compiler.

This is all available on the cloud, but there are cases you have to carry the device. Sometimes, code is large to be executed and it gets impossible to send the compiled code over internet. Also, potential threats of communication may intercept the code which is not desired. Therefore, a more comprehensive way

is to have a private cloud connected with your device and you can program it from anywhere, by switching your private cloud to public cloud.

This has importance in universities and defense. Technical enthusiasts work and program. They can easily accomplish their job. Defense has information of national importance and risking them is risking the lives of people. It has automated tanks and airplanes which can be well controlled by the cloud once we know certain setups to establish the interfacing. This is an initiative which will discover one of such methods to program device from cloud services.

4 Construction of a private cloud

4.1 Considerations for a private cloud

Various considerations were taken into account before finalizing the design for the cloud. Certain properties play their prominent roles such as

1. Scalability
2. Elasticity
3. Availabilty
4. Decrease in operation cost
5. Overall reduced cost
6. Performance
7. Security

Customers use resources as a service for e.g. storage space, computational medium, network and other resources. It indeed reduced the hustle of establishing a physical infrastructure for your own work. Also, it facilitates collaboration. Along with numerous advantages, it has certain demerits like

1. Internet connection oriented
2. High net bandwidth
3. Render dependence on data maintenance

As Cloud computing is increasing its influence throughout the world, more and more challenges are being discovered as it is a new concept in IT world. Certain challenges that are still haunting the novelty are

1. High network availability
2. Data management
3. Linearly scalable
4. Data Security e.g. confidentiality etc.

Keeping in mind all factors, first and the foremost thing was the choice of the operating system for the cloud server to develop a private cloud. Operating system is a resource manager which directly interacts with the hardware and we provide hardware and other tangible components as a service over internet. There were various online sites which help to establish the personal cloud server space like Rackspace. But our objective was to link the embedded device

in the cloud architecture to provide it as a service. Hence, deciding the OS was a necessity which could help in synchronizing the communication between components.

First of all, Microsoft products require license. Switching to open source operating systems was likely to happen because of following reasons

1. It is free. But MS products are available for hefty and recurring fees.
2. MS licenses are allowed only on a single computer but Linux distribution can be installed on any no. of computers.
3. Security aspects of linux are much stronger than of Windows.
4. One doesnt need to wait for a patch from a single owner company.
5. With linux, you have the power to control just about every aspect of the operating system.
6. So many software choices are available on linux unlike Windows compatibility issues. Softwares on linux tend to be packed with more features and greater usability than softwares in Windows.
7. Option to modify to source code along with getting them for free.
8. Linux is perfect for computers barely with any processing power and memory.
9. Disadvantages like understanding and compatibility affect little to overall problem of linux loathe.

4.2 Building of a private cloud

4.2.1 Open Source tools

Ubuntu Enterprise Cloud (UEC) consists of Eucalyptus. It is an acronym for Elastic Computing Architecture for Linking Your Programs to Useful Systems. It delivers IAAS. This tool has provision of establishing a private server and clients to that server. It has user-interface which is compatible with Amazons EC2 (Elastic Cloud Compute) and S3 (Simple Storage Service) services. Basic components are listed below [9]

Node Controller - executes actions on the physical resources that VM (virtual machine) instances.

Cluster Controller - manages a collection of node controllers (cluster) that work together.

Cloud Controller - processes incoming requests from external clients or administrators.

Walrus it is Amazon S3 API and it is used for storing VM images and user storage using S3 bucket.

Oracle Virtual Box 4.1.8 is installed. This helps to create virtual machines on host system. They allocate certain RAM and HDD to the machines created. Theses virtual machines are completely isolated from the host machine. The hard disk is stored on physical HDD as a virtual disk image (.vdi).

4.2.2 Architecture

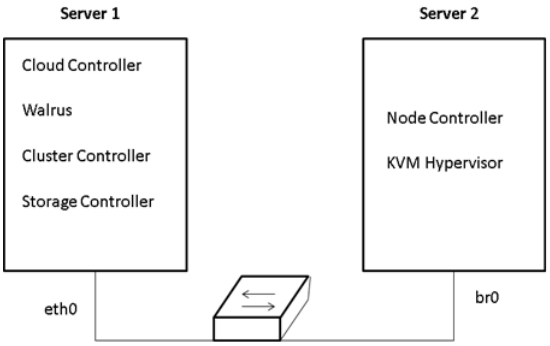

Figure 3: Implemented server architecture

This can be installed in various options. All in one machine, Node controller on one and rest on other or cloud controller, cluster controller and node controller on physically three different systems. Following the above architecture, that briefs us about the working of two servers. Cloud controller, cluster controller, storage controller and walrus are on one system, whereas node controller is on the other. Node controller is supervised by KVM hypervisor, which has the responsibility of creating virtual machine instances. It also enables live migration of images. Web UI is being accessed from the node controller only i.e. requests to cloud controller is from node controllers browser.

4.2.3 Steps Involved

After creation of two different systems, namely client and cloud, in virtual box, these steps were followed [10]

1. Install Ubuntu Enterprise Cloud with cloud controller, storage controller, walrus and cluster controller.

2. Install UEC on client machine with node controller only.

3. Edit both these machines settings, enable adapter 1 with internal network named intent option and adapter 2 with NAT.

4. Run a command on your host terminal VBoxManage dhcpserver add net-name intent ip ¡ip address of dhcp server you wish to give¿ –netmask ¡suitable mask¿ –lowerip ¡start of ip address range¿ –upperip ¡end of ip address range¿ –enable.

5. Make corresponding changes in /etc/eucalyptus/eucalyptus.conf file and /etc/network/interfaces for both the machines.

6. Set VNET public interface and private interface to "br0" and remove dhcp for br0 interface. Assign a static ip in accordance of the internal network set.

5 Proposed Model

5.1 Creation of Machine Image

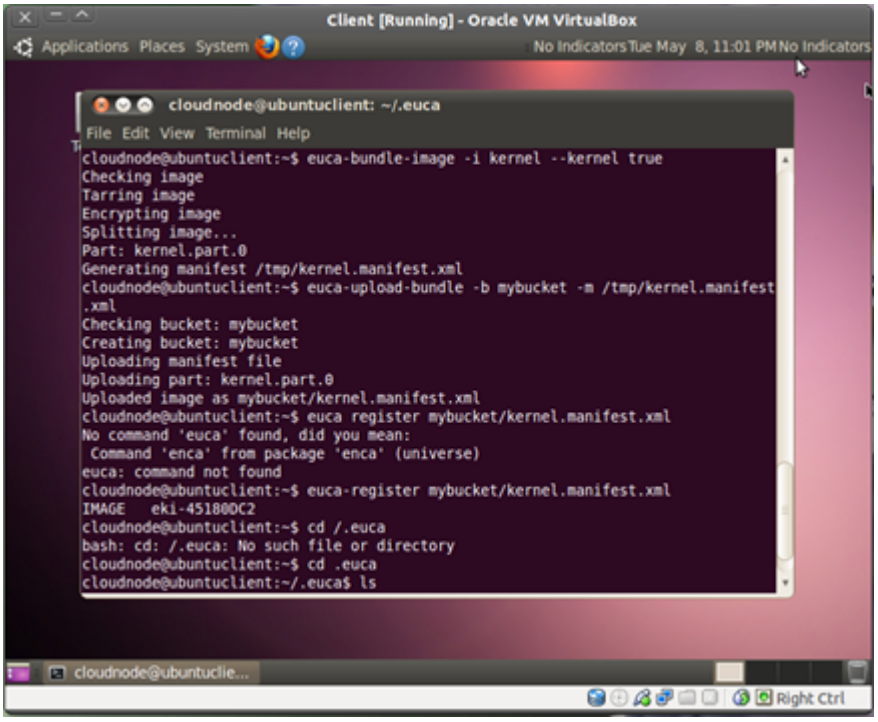

Figure 4: Showing the registration of the created kernel image

The machine image can be created and the output will be showed as above. It is created and registered on node controller. Bundling an EMI is a multi-step process involving the following [11]

1. creating a virtual disk image

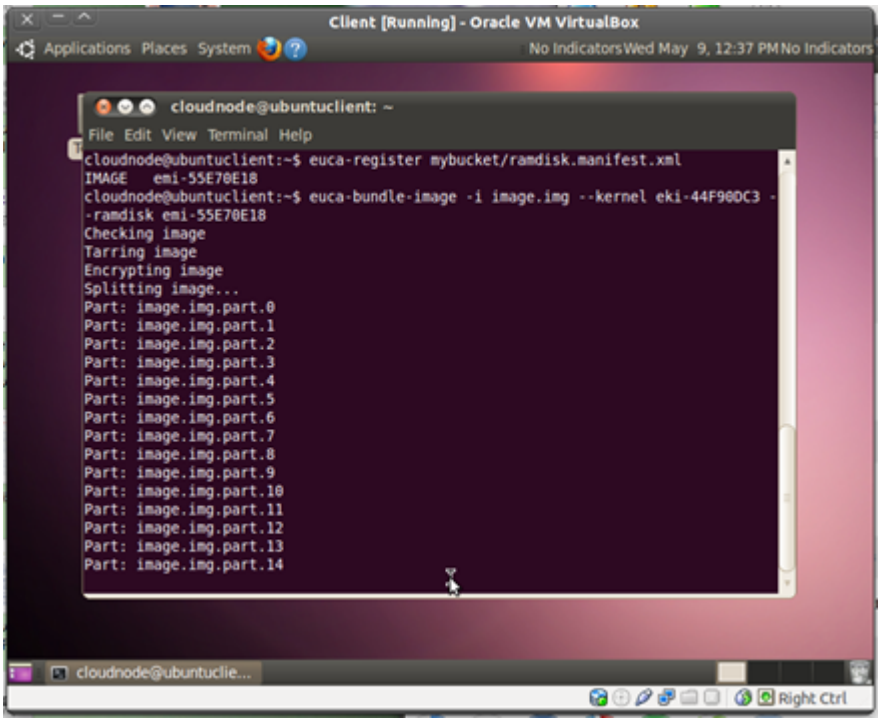

Figure 5: Creating the final registered image

2. installing the OS

3. installing required applications

4. making the OS ready to run under UEC

5. registering the images with UEC

6. testing of image

During installation, gcc4mbed is installed in the UEC instance. It is an offline compiler for programming mbed prototyping board. Once the instance starts running then the configured and uploaded packages are provided to the user,

which holds the whole idea of this project.

5.2 Proposed Cloud Architecture

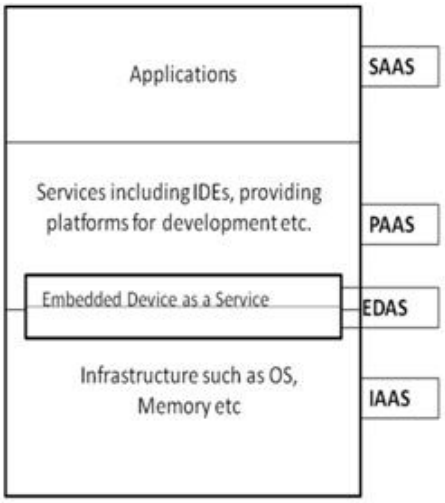

Figure 6: Proposed architecture for cloud services

IaaS has the feature of lending OS, memory, network and other physical resources. Devices can also be considered a separate physical entity. But in PaaS, you deploy your code and APIs associated will help to program the device. Hence, EDaS has a domain overlapping the both services. Using IaaS is deployed in the project but using PaaS is yet to be explored.

6 Conclusion and Future Scope

The machine image created and registered has stuck in a problem as KVM extensions refuse to work on normal hardware. Intel and AMD have certain extensions for hardware which are vmx and svm respectively. In short, current problem and there solutions are

1. KVM: it does not support LVM formatted hard dis. By avoiding LVM during installation, problem could be solved
2. USB support is not facilitated by Oracle Virtual Box a defect still to have a patch released.
3. Server and Client are connected in an internal network and hence, server is unable to use the internet connection of the host.

Along with these solutions, future scope holds the development of EDaS in overlapped domain with PaaS.

University needs to handshake new technologies opening doors for students of improvement and development. Compiling codes and programming the embedded device would be done without much ado. Keeping track of tanks equipments and airplane parts would serve the defense the greatest without any flaws. Embedded devices located within a factory or deployed remotely in mobile applications can communicate directly with cloud-based services. Devices can thus be monitored and managed from any location and collected data shared with

multiple applications which also reside in the cloud- a key consideration for industrial automation projects.

References

[1] Sun Microsystems, Introduction to Cloud Computing Architecture, 1st Edition, p.(1-2), p.(9-14), June 2009.

[2] Thomas B Winans and John Seely Brown, Cloud Computing- A Collection of Working Papers, p.(1-2), p.5, Deloitle, 2009.

[3] Joe Bungo, Embedded programming in the cloud: a novel approach to academia, January/February 2011, IEEE.

[4] Michael Barr. "Embedded Systems Glossary". Neutrino Technical Library. April 2007.

[5] Wolf, Wayne H. ,Hardware-Software Co-Design of Embedded Systems, Vol 82, no. 7, pp.968-969, July 1994, IEEE.

[6] Heath, Steve, "An embedded system is a microprocessor based system that is built to control a function or a range of functions.", 2003.

[7] Michael Barr, Anthony J. Massa, Programming embedded systems: with C and GNU development tools. O'Reilly. pp. 12, 2006.

[8] Mbed Hardware Specification, http://mbed.org/handbook/mbed-Microcontrollers. Tutorial.UG682(v1.0)April 27,2009

[9] Ubuntu Enterprise Cloud Installation guide,https://help.ubuntu.com/community/UEC/CDInstall.

[10] Ubuntu Enterprise Cloud Topologies, https://help.ubuntu.com/community/UEC/Topologies.

[11] Eucalyptus Beginners Guide, Image Management, http://cssoss.wordpress.com/2010/05/10/eucalyptus-beginner