

Design, Implementation and Evaluation of Unified Communications on-premises and over the Cloud

Aklilu Daniel Tesfamicael, Vicky Liu and William Caelli

Science and Engineering Faculty
Queensland University of Technology
Brisbane, Australia

aki.tesfamicael@hdr.qut.edu.au, {vicky.liu, w.caelli}@qut.edu.au

Abstract

Unified Communication (UC) is the integration of two or more real time communication systems into one platform. Integrating core communication systems into one overall enterprise level system delivers more than just cost saving. These real-time interactive communication services and applications over Internet Protocol (IP) have become critical in boosting employee accessibility and efficiency, improving customer support and fostering business agility. However, some small and medium-sized businesses (SMBs) are far from implementing this solution due to the high cost of initial deployment and ongoing support. Cloud based UC solution, UC as a Service (UCaaS), is now itself a maturing technology in the marketplace and it has revolutionized the IT industry, being the powerful platform that many businesses are choosing to migrate their on-premises UC solution onto. UCaaS solution has the potential to reduce the capital and operational expenses associated with deploying UC on their own. In this paper, we will discuss and demonstrate an open source on-premises UC solution, viz. “Asterisk” for use by businesses, and report on some performance tests using SIPp. This paper also discusses and demonstrates an open source UCaaS solution. The contribution from this research is the provision of technical advice to businesses in deploying UC and UCaaS, which is manageable in terms of cost, ease of deployment and support.

Keywords: Unified Communications (UC); Unified Communications (UCaaS); Voice Over IP (VoIP); Session Initiation Protocol (SIP); Infrastructure as a Service (IaaS); Private Cloud; Hybrid Cloud

1. Introduction

In recent years we have seen many collaborative software and related tools in the market aimed at giving business confidence to expand. In this fast moving world of technology, businesses expect immediate access to these tools for their employees from a wide range of communication devices. This access allows employees to get connected from anywhere at any time provided they have internet access.

We have started to see the evolution of UC in the Information and Communication Technology (ICT) industry in recent years. Within ICT a definition of UC is still debatable but many define UC as the integration of real-time communication services or systems. For many, such communication services include “voice over internet protocol” (VoIP), video conferencing, instant messaging (IM), presence, and voicemail. An increasing number of Government and Private Business Enterprises are deploying UC for first time or replacing their old telecommunications and like systems to eliminate redundant equipment and to be able to converge and integrate all forms of communications systems under one platform. These features have been shown to improve productivity and efficiency by enabling consumers and businesses to communicate effectively in the 21st century. However, the efficient and reliable deployment of such services or projects requires unique skills. Most

businesses engage third party companies (vendors) who specialize in UC to do the work for them and some do it in house by hiring contractors who have extensive skills in Unified Communication engineering. Moreover, leading providers of UC products and systems, such as Cisco, Microsoft and Avaya [1], are pushing their products to the market with associated high prices which have become too steep for the small and medium sized business (SMBs).

To save operational costs and improve productivity many businesses especially SMBs are now moving towards a cloud solution for their UC. The transition to cloud is quite simple for those that have no UC deployed on-premises. However, it is highly advisable to carefully plan for transitioning to UCaaS from exiting on-premises UC solution. UCaaS can be regarded as a cost-effective model for on-demand delivery of unified communications services in the cloud. Cloud computing gives businesses on-demand, fast access to a variety of IT services, software and applications while also giving businesses a shared pool of configurable computing resources at the platform, infrastructure and application layers. Thus cloud technology helps businesses do more and act faster by giving them the option to select IT services without having to fully fund, build, manage or maintain them. SMBs would prefer a “pay as you go” method for funding ICT services in the cloud as this is cheaper for them. However, some large organizations may prefer to build their own private, public or hybrid cloud environment.

In recent years, the need for movement to a cloud solution environment has gained more attention and as a result most organizations are looking onto ways of moving into cloud. Recently, the Queensland Government in Australia has announced their ICT strategy to make Queensland Government a “cloud-first” enterprise [2]. This will require proper planning going forward and decisions on the selection of cloud providers so that they can successfully migrate data centers to IaaS, “infrastructure as a service”, and their IT services and applications such as UC to equivalent services in the cloud. However, to boost business confidence in this area there is a need for more research so that the customers may have a clear understanding of what cloud technology can provide to them and under what trust environments.

One of the Authors has been working as a UC Engineer and Solution Architect for number of years and has been involved in many government and private business enterprise UC projects including transitioning to Cloud. It is understood that such activities require a business to have adequate funding for implementation of associated UC products and systems such as hardware, software, licenses, deployment and ongoing maintenance. However, these costs can be reduced by selecting the Cloud solution. This may seems to be reasonable for larger businesses but not for many SMBs. This paper, discusses the implementation and performance analysis of an Open Source UC solution based on the open source product “Asterisk” in on-premises and also discuss the deployment of UC in the Cloud. The proposed solution is believed to be cost effective and requires fewer people to manage and support. It is anticipated that the proposed solution will give confidence to businesses in moving towards a UC solution with benefits from the features of UC to help improve their productivity and boost employee confidence.

The remainder of the paper is organized as follows. Section II includes related works. Section III examines two real case scenarios. Section IV details the proposed UC solution on-premises. Section V gives a description of the experimental lab used followed by an analysis of results in Section VI. Section VII discusses proposed UC solution in the Cloud followed by conclusions and proposals for future work in Section VIII.

2. Related Work

In the last few years, numerous studies on open source VoIP have appeared. A large portion of the literature on “Asterisk” reveals its focus on VoIP. Many researches do not appear to take an holistic approach to studying unified communications by including the

other features of UC such as voice messaging, conferencing and mobility. Gohel and Lakhtaria [3] have presented some considerations in relation to a theoretical implementation of an Asterisk server to provide a VoIP system but the report lacks any assessment of an appropriate test of the system. Implementation of a VoIP service without proper performance testing has a high risk of system crashes or inferior call quality. Ahmed and Mansor [4] report results of practical experiments accomplished on CPU dimensioning and the performance of an Asterisk VoIP PBX are given. They used the SIPp simulator to generate voice calls to measure the performance of the Asterisk CPU. Their results found that incidence of a CPU spike increases when concurrent calls are incremented. It is true that the overall performance of Asterisk will generally be affected by the need for large calculations and thus it is essential to select a computer with a powerful Central Processing Unit (CPU)/Floating Point Unit (FPU). However, in our opinion, we also suggest that other tests need to be carried out including overall network performance, bandwidth tests for geographically diverse VoIP deployments and Quality of Service (QoS). Konstantoulakis and Sloman [5] have demonstrated successful implementation of a call management policy specific to an “Asterisk” IP PBX. The study looked at implementation of a graphical user interface for ease of Asterisk system administration. Hammoud and Bourget [6] observed the integration of the Asterisk server with a rule based engine (InRule) to enable the Asterisk Server to instruct “InRule” to perform the required analysis. In a similar manner Chava and Ilow [7] reported on the successful integration of an Asterisk server with the Cisco Call Manager for interoperability.

There are also numerous studies focused on the open source cloud solution. However, at the time of writing this paper and during our literature review we have not found any published paper that discuss UCaaS or Voice over IP as a service (VoIPaaS) in an open source cloud platform. As such, this literature review is mainly focused on the type of cloud platform where our experimental UC is deployed in the Cloud.

Xiaolon et al. [8] performed a study comparing two popular types of cloud, open source platforms, viz. “Openstack” and “OpenNebula”. They examined these two products from different angles in comparing their provenance, architecture, hypervisors and security. This study may influence further study in the area. However, there is no experimental evidence to verify their proposals. It would have been an ideal solution if both Openstack and OpenNebula have been implemented in virtual or dedicated physical hosts to provide experimental performance results to support their proposals.

Nasim and Kassler [9] deployed Openstack in a virtual infrastructure and dedicated hardware to observe performance parameters. The result from this type of study can support decision making when selecting the underlined hardware in deploying Openstack. Though this solution was deployed on a small scale the paper demonstrated Openstack deployed over dedicated hardware always performs better than Openstack running over a virtualized environment due to overheads from computational resource usages.

There are other studies [10-15] that focus on architecture designs and Openstack performance analysis and these have indicated that there is a great interest in this area.

Rosado and Bernardino [16] have produced a paper discussing Openstack Architecture. However, there is no original work produced in this paper.

In recent years’ researchers have delivered major research papers on open source cloud solutions that are promising. However, there is a need for more research beyond consideration of any underlying infrastructure to understand the deployment and performance of applications and other services hosted on the underlying IaaS platform. This paper is different from those reviewed as it has mainly focused on the implementation of the underlying architecture of Openstack, that forms the base for “Infrastructure as a Service” (IaaS) and the deployment of cloud services such as UCaaS, hosted in that open source cloud environment.

3. Real Case Scenarios

A. Senario 1

Many large business are looking at how they can deploy unified communications and how they can do so cost effectively. Some companies are well organized at rolling out UC but some have no plan regarding how they can deploy it.

In this scenario we take a look at one of the large scale energy companies in Australia, where one of the authors of this paper was part of a technical team in deploying UC globally. The company has more than 20,000 employees globally.

The project team was tasked with defining what the Unified Communications technology implementation would look like for the stated company based on the premise that the company already had some core communication projects underway. However, they lacked a holistic view on how to bring these together to enable full UC capabilities.

Their project started with 78 problem statements which were grouped into 8 capability areas within the company unified communications framework. By overlaying strategic product, vendor constraints and market research from Gartner and Forrester it was able to identify a number of integration areas where there were multiple solution options.

For each of the option areas identified, vendor workshops with Microsoft and Cisco were conducted to confirm that the proposed options would satisfy the original end user problem statements. This enabled refinement and further analysis of the proposed options; and further consultation with industry analysts.

The project team comprises of a UC Architect, UC Engineers, Project Manager, Stakeholders and Vendors. The approach taken to deliver this project had five stages: Inception, Definition, Building, Deployment and Transition. The recommendation given to the company was to move fully to the Microsoft "Lync" enterprise level solution. The first step was to integrate their existing Cisco Unified Communication Manager (CUCM) which was providing VoIP, but not UC as such, with Microsoft Lync which provides fully functioning UC using a SIP Trunk. During the integration and migration phases, the project encountered many challenges. The technical team who had expertise in Cisco systems lacked in-depth knowledge of the Microsoft system. Moreover, the resources for deploying this UC are even a costly exercise for large scale enterprises.

All Cisco hardware and software has to be replaced by Microsoft Lync certified hardware and software platform. The cost of the team involved for deployment and management is also an expensive consideration.

B. Scenario 2

One of the Authors of this paper currently works with government departments and is responsible for design, implementation, integration and support of Unified Communications. One year ago the department initiated a project to integrate their existing Avaya telephone system with Microsoft Lync to obtain benefits from video conferencing, messaging and presence, as the existing configuration and version of the Avaya System does not provide full UC functionality. The total cost of ownership, including ongoing maintenance and license costs, is very expensive. However, recently the department decided to move to Cloud based UC solution (UCaaS). This exercise has become very expensive as the business has already invested in integrating their legacy telephone system with Lync. The lack of a technology roadmap, a plan that matches the current and future technology deployments, with the aim to meet short and long term goals is affecting businesses with unnecessary operational and capital expenditures in deploying UC on-premises and transitioning to Cloud based solution.

C. Scenarios Analysis

What we have learned from these two real case scenarios is that the deployment of UC and/or transitioning to Cloud UC solution requires a proper plan with the skilled UC implementation profession with adequate funding. The requirement and scale of UC deployments may vary from business to business. However, the two real case study scenarios are based on large scale enterprises and may not be adequate for SMBs. Moreover, the UC solution provided by leading vendors such as Cisco, Microsoft and Avaya [1] are expensive. Therefore, this research proposes an alternative implementation for businesses realizable on an open source platform. We have also learned from scenario 2 that the transition to cloud requires adequate planning to avoid a high cost associated with transitioning.

4. UC Deployment on-premises

D. Our Proposed Implementation

We have chosen “Asterisk”, the open source IP PBX, for our proposed solution due to its support of fully unified communications and a wide range of VoIP protocols including H.323, the Media Gateway Control Protocol (MGCP), and Session Initiation Protocol (SIP). The “Digium Asterisk” is by far the most mature and popular open source IP PBX currently available in the market. It includes all the building blocks needed to create an IP PBX, an Interactive Voice Response (IVR) system, a conference bridge and other related communications services.

E. UC Overview

We have defined UC under six specific modules as used in our UC implementation:

- Presence
- Instant Messaging and Application Sharing
- Conferencing (Voice, Video, Web Conferencing and Application Sharing)
- Email and Voice Messaging
- Voice over IP (VoIP)
- Mobility

It has been shown, through numerous case studies from different enterprises who have deployed UC, that costs associated with overall enterprise communications are greatly reduced. Employees have more communications options. The mobility of UC allows employees to communicate remotely.

The Asterisk server is supported on a VMware virtualization platform as well as on a physical server/workstation. In our experiment we installed Asterisk on a VMware and physical workstation.

Our high level proposed implementation is illustrated in Figure 1.

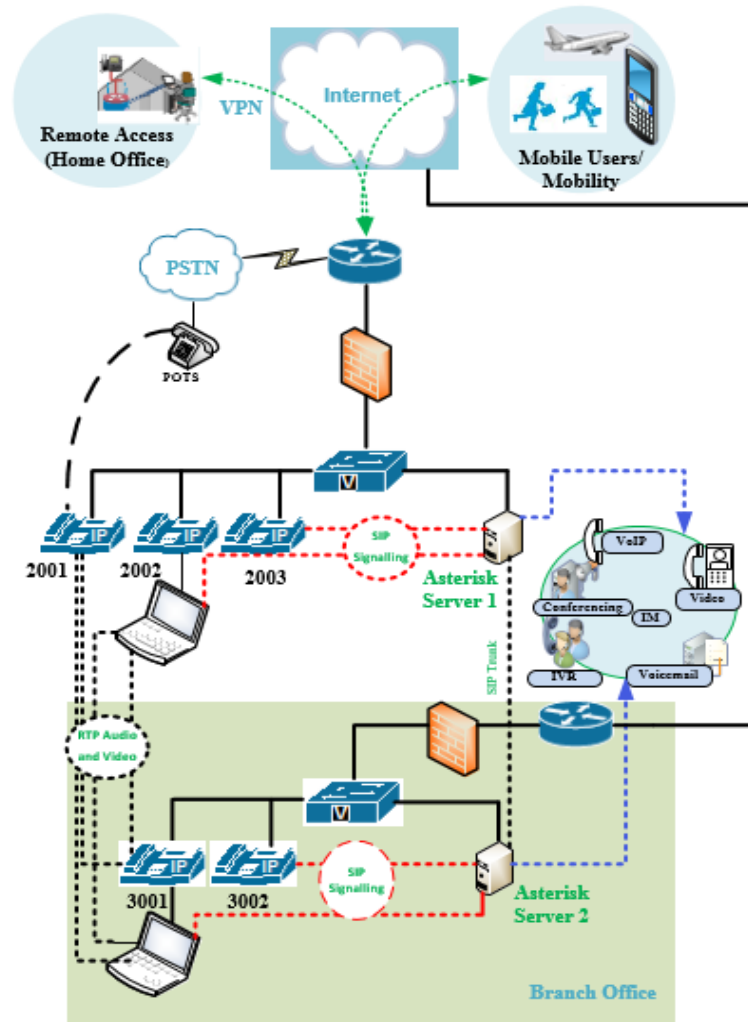


Figure 1. High Level Design – Proposed Design Solution

To facilitate Internet connectivity one connection to an Internet Service Provider (ISP) is required. An Integrated Services for Digital Network (ISDN) service option is also required for digital transmission of voice and video outside the business network. We recommend the connection of one E1 service or the use of SIP connect to eliminate or minimize the need for a costly gateway.

F. UC Protocols

Most common UC deployments with Asterisk use SIP, H.323 and IAX UC protocols. However, SIP is the most widely used protocol. Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences. The “SIP.conf” part of an Asterisk setting is used to configure the default settings used for SIP calls. This SIP configuration is a core part of the Asterisk server as most of the calls is sent using SIP. We recommend the use of a SIP trunk to connect two Asterisk servers located in different locations. SIP registration and session establishment is presented in Figure 2.

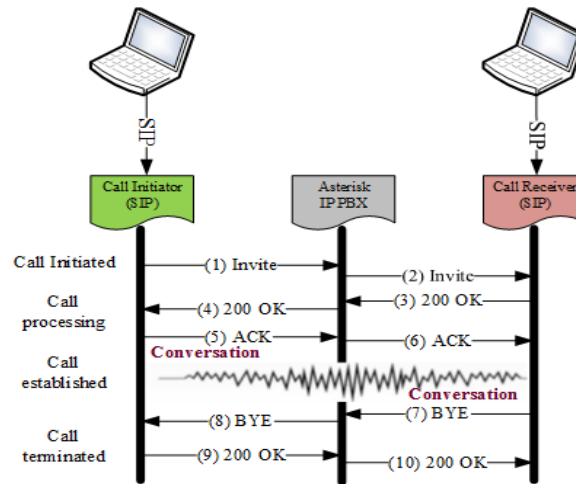


Figure 2. SIP Session Signal Flow Diagram

We recommend the use of “FreePBX”, an open source graphical user interface (GUI) to modify the configuration file. This graphical interface controls and manages an Asterisk system which makes building and managing of the Asterisk system easier. FreePBX is a full-featured PBX web application.

In Asterisk, channels are created using Channel Drivers and call processing is centered around those channel drivers which handle all the protocol-specific details of ISDN, SIP, and other telephony protocols and communicate with Asterisk. Figure 3 illustrates the integration of our two Asterisk servers using the SIP trunk. The “rtp.conf” file in Asterisk controls the Real-time Transport Protocol (RTP) ports that Asterisk uses to generate and receive RTP traffic. The RTP protocol is used by SIP.

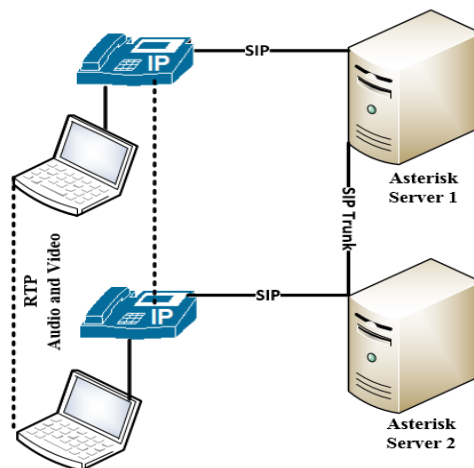


Figure 3. RTP Audio and Video Flow Diagram

5. Description of the Experiment

To demonstrate the feasibility of the proposed implementation we setup two Asterisk servers connected via an Integrated Services Router, a “Cisco 2811” unit. We have also setup a SIPp client for stress testing of the Asterisk server. This task was conducted by post graduate students undertaking a “24 credit point” project who setup the Asterisk system based on the implementation guidelines provided in this paper. The students had no prior knowledge and experience in Unified Communication. However, they were able to implement and successfully manage the project.

For this deployment the two Asterisk servers are setup in the Networking Lab at the Queensland University of Technology's (QUT) Gardens Point Campus. For SMBs, both these servers could be located at different sites, one being in a head office and the other one at one of the branch sites, depending on the size of the business. If the size of the business is small we recommend the use of one Asterisk server. However, the business can still benefit from the six modules we have discussed above.

The purpose of this stress test is to identify the performance threshold of Asterisk in a real world deployment scenario.

Figure 4 shows the high level design of the test lab we implemented.

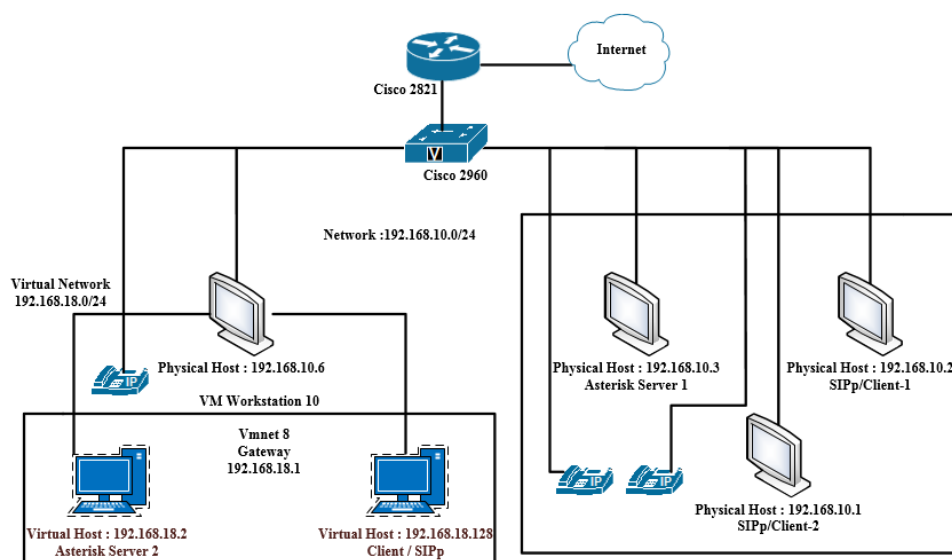


Figure 4. Test Bed High Level Design

The hardware specifications are listed in Table 1.

Table 1. Test Lab Design Hardware Specifications

4 HOSTS	1 ROUTER 1 SWITCH
Asterisk1: Intel Core i7-3770 CPU @ 3.40GHz x 8, 16GB Memory (RAM) and 235 GB disk size. Operating System (OS) type CentOS 6.5 Asterisk2: Intel Core i7-3770 CPU @ 3.40GHz, 8GB memory (RAM) and 458 GB disk size OS type: CentOS 6.5 SIPp Client 1: Intel Core i7-3770 CPU @ 3.40GHz, 1GB memory (RAM) and 10GB disk size Operating System (OS) type: Ubuntu 14.04 LTS (64-bit) SIPp Client 2: Intel Core i7-3770 CPU @ 3.40GHz, 1GB memory (RAM) and 10GB disk size Operating System (OS) type: Ubuntu 14.04 LTS (64-bit)	Router: Cisco 2821 Router Switch: Cisco 2960 Switch

6. Analysis of the Experimental Result

As a first step, before we proceeded to the system performance tests, we performed a full UC operational test based on the modules we have defined in this paper.

For Instant Messaging (IM) and Presence to work in Asterisk we installed "OpenFire", a real time collaboration (RTC) server and for a client we installed "Spark", an Open Source cross-platform IM client.

G. Voice and Voice Call Test

Figure 5 shows a phone call conversation between two IP soft-phones based on the G.711 standard.

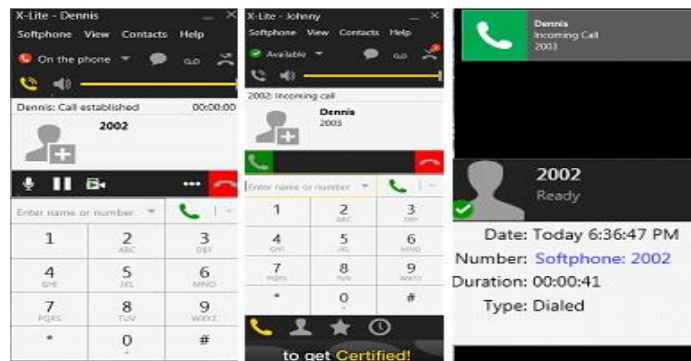


Figure 5. Voice to Voice Call using G.711 Codec

Figure 6 shows a video call conversation between two IP softphones based on the G.711 standard.

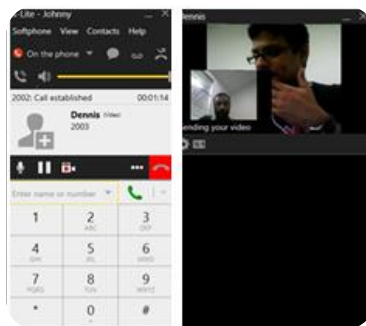


Figure 6. Video to Video Call using G.711 Codec

H. Performance Evaluation

We analyzed the performance of the Asterisk server under a variety of scenarios.

To test SIP performance against Asterisk PBX call quality, there are several software tools available such as SIPp, a free open source test tool, and traffic generator for the SIP protocol. The benefit of SIPp is the ability to create simultaneous connections in order to test the performance of the Asterisk Server.

In our lab we made few configuration changes for files, “sip.conf” and “extension.conf”, on the Asterisk Server to suit our needs. Table 2 shows the configuration of “sip.conf” and “extension.conf” after we made the changes.

Table 2. “sip.conf” and “extension.conf” Configurations

[SIP.conf]	[Extension.conf]
type=friend context=sip host=dynamic port=6000 user=sip canreinvite=no disallow=all allow=ulaw	exten => 3001,1,Answer() exten => 3001,2,SetMusicOnHold(default) exten => 3001,3,WaitMusicOnHold(20) exten => 3001,4,Hangup()

SIPp can perform server stress tests using different commands.

sipp -sn uac -d "calls in milliseconds" -s "extension number" "IP address of a server"-l "number of simultaneous calls"

That is, this command generates sip invites as a client with duration of calls in milliseconds, dials the IP address of our UC server, and reaches the extension number with a limit to the given number of simultaneous calls.

For our purpose we run the following command to generate calls for the test bed.

sipp -sn uac -d 20000 -s 3001 192.168.10.3 -l 30

This command connects as a client, gives the duration of the call 20 seconds, dials the IP address of the Asterisk server 192.168.10.3, and reaches the extension 3001, with a limit of 30 simultaneous calls.

Initially we started the test by running 30 concurrent calls and the Asterisk server CPU usage went up by only 12.9% using G.711 codec. We incremented the number of concurrent calls by 20 and the CPU usage increased by just a further 5%. We kept on testing until we reached 500 concurrent calls at which time the processor usage of the Asterisk server reached 93.1%. We have also followed the same process to test CPU performance of the Asterisk server using the Transcoding GSM-G.711 codec to observe the CPU performance during Transcoding. Figure 7 summarizes the comparison of results revealed using G. 711 and Transcoding.

The experiment has been repeated three times and the results revealed in this paper are the mean (average) value of the three measurements, rounded to the nearest whole number

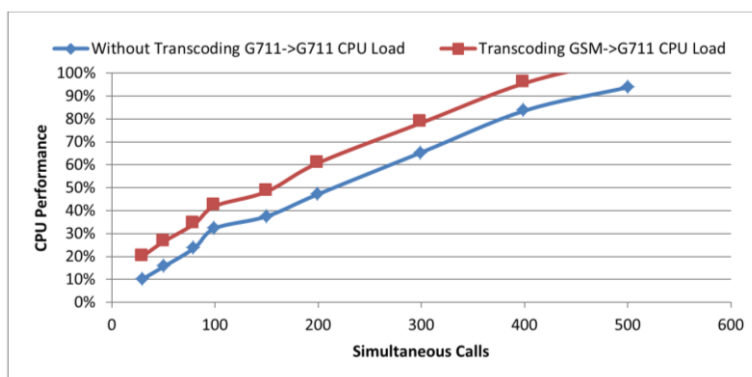


Figure 7. Intel Core i7 3.40GHz CPU Performance on G711 and Transcoding

As Asterisk supports various codecs we have also performed some comparison in CPU usage based on G.711 and G.729 codecs as shown in Figure 8.

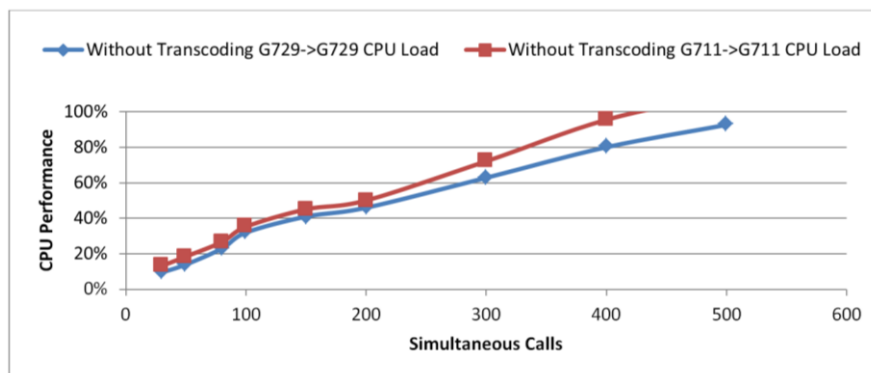


Figure 8. Intel Core i7 3.40GHz CPU Performance on G729 and G.711

From our experiment we observed that G.729 consumes relatively fewer CPU resources than the G.711 codec.

What we can conclude from this experiment is that the Asterisk Server, Intel Core i7-3770 CPU @ 3.40GHz, we used for this experiment can handle about 300 simultaneous calls (CPU load about 65%) using G.711. If we use transcoding GSM->G.711 it can handle about 200 simultaneous calls. However, for areas with limited bandwidth the use of the G.729 codec is recommended as the CPU performance is slightly better than with the G.711 codec.

7. UC Deployment in the Cloud

I. Openstack Architecture

Openstack is an open source cloud computing project aimed at deployment in all types of cloud environments. Globally, cloud computing experts contribute to this project to make its implementation simple and scalable [19]. Openstack provide such IaaS solution through different forms of services. The simplified architecture of Openstack, and its components that form IaaS, is shown in Figure 9.

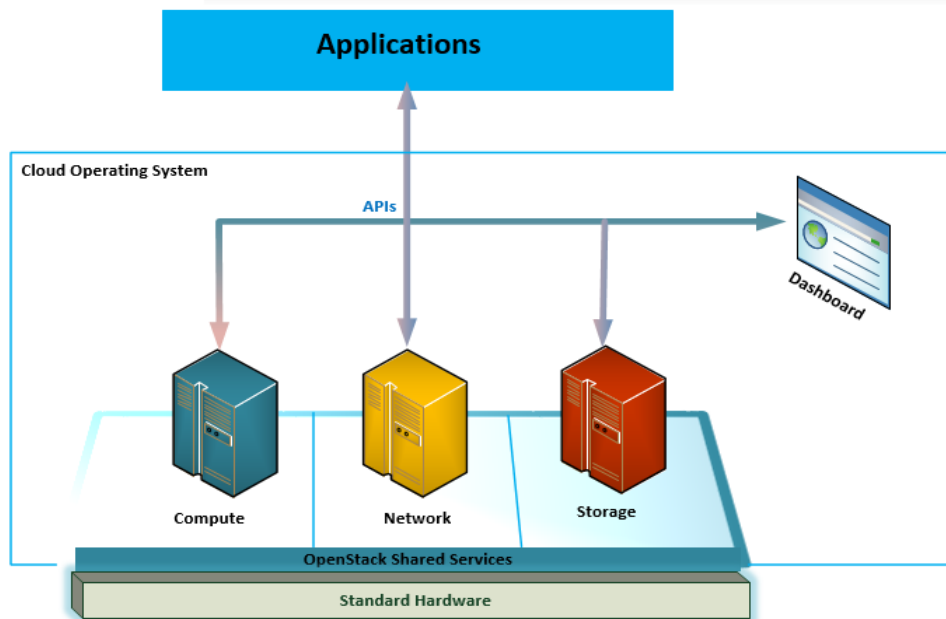


Figure 9. Openstack Architecture

The main components of the openstack cloud model are controller, compute and network nodes.

- Controller Node hosts all Openstack services needed to orchestrate virtual machines deployed on the Compute Nodes [17]. For high availability it is recommended to deploy at least three controllers to provide multiple controller nodes.
- Compute Node is, in many ways, the foundation of the environment; it is the server on which virtual hosts are created and applications are hosted. Compute nodes need to communicate with controller nodes and access essential services such as MySQL.
- An Openstack environment includes multiple servers that need to communicate with each other and to the outside world and the Network Node serves that purpose. It supports both old “nova-network” and new “neutron” based Openstack networking implementations

Many proprietary cloud service providers have already started to show interest in the Openstack solution. For example, Cisco Inc. has recently managed to integrate emerging Openstack cloud solution with their cisco unified computing system (Cisco UCS) and Nexus platforms [18].

J. Our Proposed Implementation

The deployment of UC on-premises is becoming more complicated and expensive [19]. To avoid this, businesses are looking to the cloud as an outsourcing option. UCaaS gives businesses quick access to current business collaboration tools. UC service providers often offer multi-tenant cloud services. This UC model is appropriate for smaller or medium business (SMBs) because of its flexible pricing. Large businesses that require their own cloud environment may benefit from the proposal outlined here better than SMBs. Cloud service providers may also look on this proposal as an alternative solution for their proprietary cloud offerings. Our high level IaaS and UCaaS design overview is shown in Figure 10.

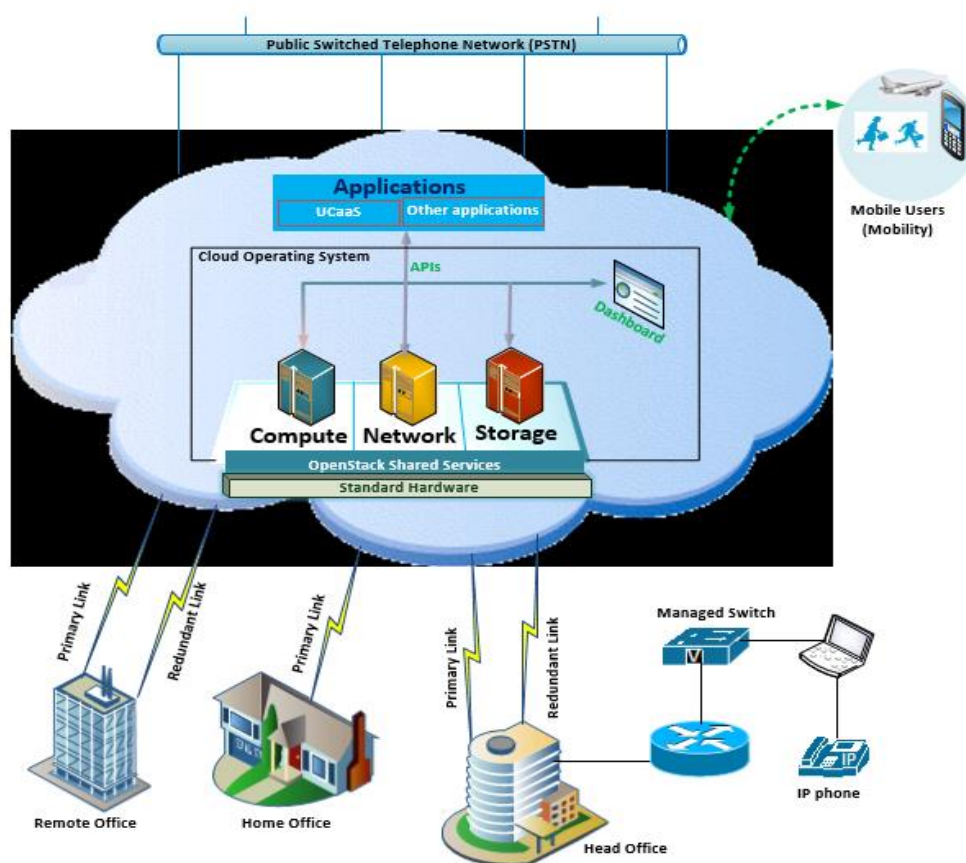


Figure 10. Our IaaS and UCaaS High Level Design Overview

We have used “Asterisk” unified communication software for our experiment, which is installed on a cloud virtualized host within Openstack. This Asterisk software provides voice over IP as a service (VoIPaaS), video conferencing as a service (VaaS) and messaging as a service (MaaS). All UCaaS features over the cloud profoundly depend on quality of service (QoS), storage and streaming protocols. Though this experiment focuses on a design and deployment discussion of these services in the cloud, in future work we will be testing the QoS and performance aspects of these hosts over the cloud environment, which is the vital part in considering hosting UC in the cloud.

Our Private Cloud and UCaaS Implementation Test Bed

This section describes Openstack and UCaaS implementations with dedicated hardware for the implementation of IaaS and the deployment of UCaaS to be hosted as a cloud service. For ease of deployment we have deployed Openstack in one computer, all-in-one, acting as controller, network and compute nodes as described in Table III. Openstack can also be deployed in a virtual environment [9]. For a single-machine installation, the following minimum system specifications are recommended.

- CPU that supports VT-x/AMD-V. Quad-core or better
- 12GB memory or above
- 100GB of hard drive or more
- Juju version 1.18.3 or higher
- Ubuntu 14.04 LTS, 64-bit version

In real world deployments we highly recommend this design proposal to be deployed in a multi-server solution to avoid a high potential for single point failure and performance bottlenecks. In future work we will be examining the deployment of the Openstack cloud in a multi-server platform to analyse the QoS and perform system performances tests

Table 3. Hardware Specifications of Experimental Devices

2 Hosts	1 Switch	1 Router
Host 1 – All-in-One Openstack Processor: Intel Core i7-3770 CPU, 3.40 GHz x 8 Installed memory (RAM): 16 GB Storage: 500GB OS type: Ubuntu 14.04 LTS 64bit	Cisco Catalyst 3560 Switch	Cisco 2911 Integrated Services Router
Host 2 – Management Processor: Core 2 duo, 2.33 GHz Installed memory (RAM): 4 GB Storage: 80GB		

K. Openstack Deployment Components

In our test bed we implemented three core components of Infrastructure as a service (IaaS) as discussed above. These resources are integrated in a cloud environment to deliver cloud services. The cloud deployment under this model can be delivered in an "all-in-one" host (installed in a virtual or physical host) or in a multi host configuration.

L. Description and Analysis of the Experimental Design

To demonstrate the feasibility of the proposed implementation of our private cloud solution using Openstack we setup an all-in-one server with dual network interfaces (NIC) connected to the internet via an integrated service router, a "Cisco 2821" unit. This server is running Ubuntu 14.04 TLS 64bit as an operating system. We have also used "juju" which is a powerful GUI and command line tool that helps configure, manage, maintain and deploy an Openstack cloud using "charms". Juju charms define how services are connected to each other, scaled and deployed and they also contain configuration settings of an application. Juju admin GUI for management of the cloud services is shown in Figure 11.

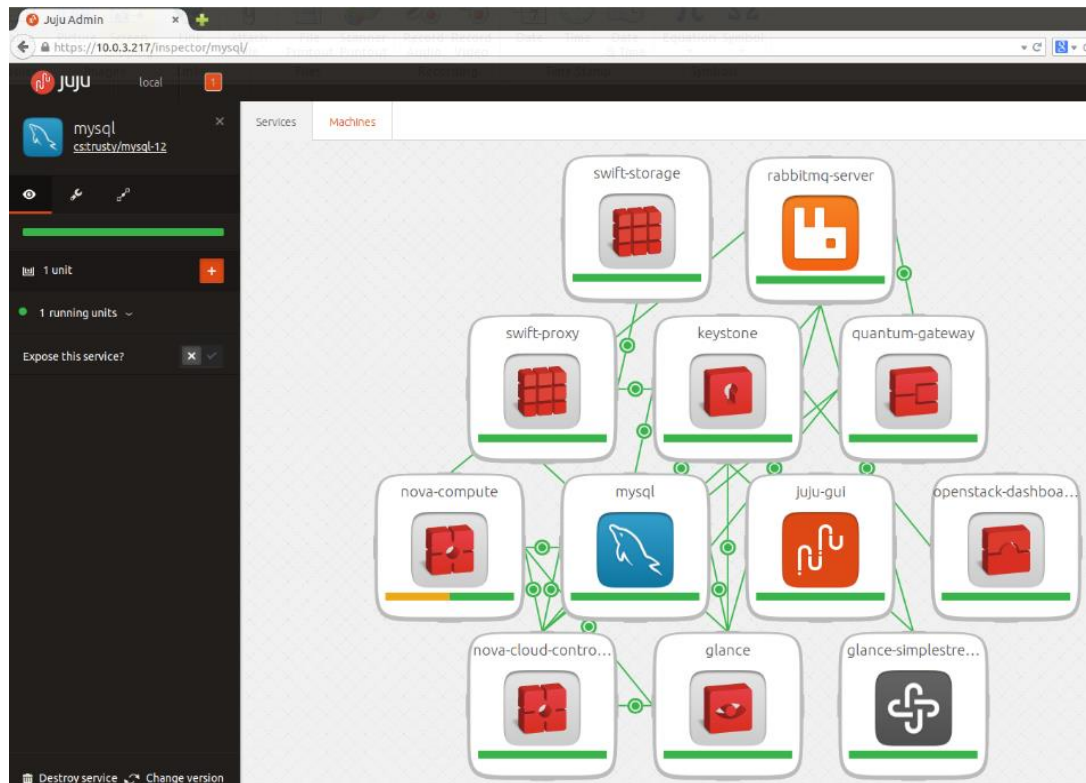


Figure 11. Our Juju Admin GUI Overview

The juju Openstack package deploys the core open cloud hosts: controller, network and compute nodes. Using the compute node we created a new virtual host and deployed cloud version “Asterisk” package on this host to serve as UCaaS as shown in Figure 12.

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Uptime	Actions
Asterisk	Asterisk	10.0.3.206	m1.medium 4GB RAM 2 VCPU 40.0GB Disk	-	Build	nova	Spawning	No State	0 minutes	Associate Floating IP More

Figure 12. Our Deployment of Asterisk in Openstack

The single machine is capable of holding the entire stack in a single LXC container, also known as a Linux container. The deployment within the stack allows us a quick test of a new Openstack deployment. On a different network, external to the cloud environment, we also installed and configured X-Lite, SIP- based UC, as softphones in two of our laptops at a remote site. These softphones are configured and registered with the Asterisk server in the cloud environment. We have successfully demonstrated a voice and video call, voice message delivery as well as the exchange of instant messaging between the two laptops. Figure 12 shows the successful voice to voice call over the cloud (VoIPaaS) between the two softphones installed on separate laptops in the remote office.

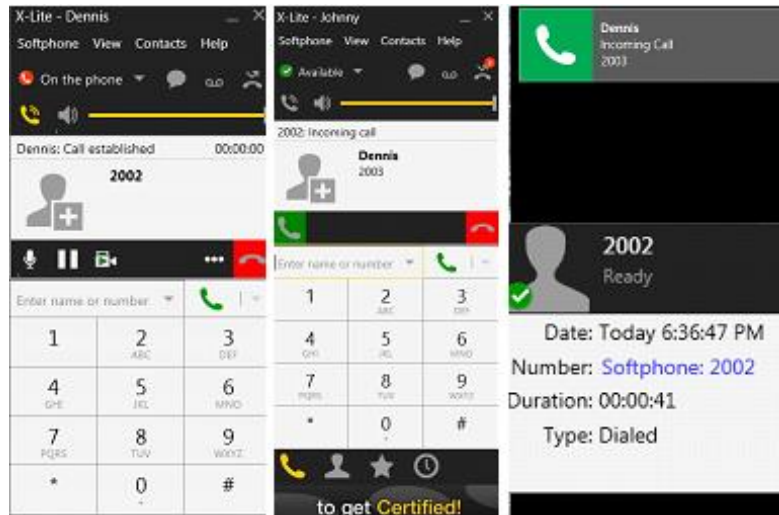


Figure 13. Our Voice to Voice Call over the Cloud (VoIPaaS)

Similarly we have managed to make a video call between the two laptops with X-Lite installed on them.

Figure 14 shows the successful video to video call over the cloud (VaaS) between the two softphones installed on separate laptops in the remote office.

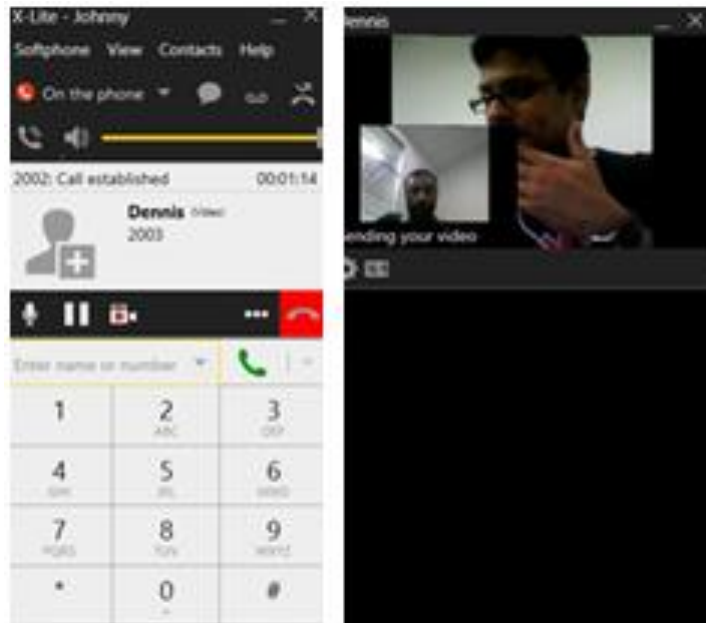


Figure 14. Our Video to Video Call over the Cloud (VaaS)

This test demonstrated the successful implementation of IaaS using an opensource cloud solution called Openstack and UCaaS using the Asterisk server. We have also successfully deployed softphones in the remote office connected directly to Asterisk server in the cloud.

8. Conclusion and Future Work

The implementation guidelines we are proposing for on-premises is manageable in terms of cost, ease of deployment and support. Anyone with fundamental data networking skills can manage to get the basic Asterisk system up and running. However, dial-plan configuration and full support for the system require advanced knowledge of the system.

We recommend business involve a UC professional to provide technical system support or to engage the vendor or UC service provider under a support level agreement.

The design and implementation in the cloud we have discussed in this paper is easy to deploy and the cost associated with its implementation is greatly reduced due to its “free to use” software license. It also avoids proprietary lock-ins from product vendors. One of our recommendations, which is based on the experimental experience, is that the deployment and configuration of this open source cloud solution can be achieved by people with basic knowledge or experience of cloud technologies.

We submit that this research paper provides useful advice to businesses that want to build their own on-premises or cloud environment and host their own services and applications. We also believe this contributes to knowledge for cloud service providers to understand the alternative solutions to proprietary cloud products, systems and services. The significance of this research is the provision of detailed technical advice and guidance to those involved in planning, designing and managing unified communication in the cloud as well as on premises, viz. in public, hybrid and private cloud structures.

In future work we will be studying the deployment of Openstack on a multi-server platform and investigate enhanced security architectures for UC positioning in a cloud environment. We will also providing a performance analysis of the core open source cloud hosts and the QoS of the dependent network in delivering voice and video services over the cloud.

Acknowledgment

Our UC implementation and performance testing on-premises has been conducted on a networking laboratory with student-formed teams. The authors appreciate the contributions from Jiexuan Chen, Fabien Chastel and Dennis Pramod. The authors also appreciate the contribution from Daniel Huang and Wang Shuai on conducting the Openstack installation.

References

- [1] M. Parker, “Gartner 2013 UC Magic Quadrant: Maturity, Challenges, Directions for the Future [Online]”, Available: <http://www.ucstrategies.com/unified-communications-strategies-views/gartner-2013-uc-magic-quadrant-maturity-challenges-directions-for-the-future.aspx/>, (2013) August.
- [2] Queensland Government, “Cloud computing Strategy [online]”, Available: <http://www.qgcio.qld.gov.au/initiatives/cloud-computing>, (2014) May.
- [3] C. K. Gohel and K. I. Lakhtaria, “Implement VoIP Based IP Telephony with Open Source Asterisk Architecture”, International Journal of Interdisciplinary Telecommunications and Networking, vol. 2, (2010), pp. 1-11.
- [4] M. Ahmed and A. M. Mansor, “CPU dimensioning on performance of asterisk VoIP PBX”, in 11th Communications and Networking Simulation Symposium, CNS’08, April 14, 2008 - April 17, 2008, Ottawa, ON, Canada, (2008), pp. 139-146.
- [5] G. Konstantoulakis and M. Sloman, “Call management policy specification for the asterisk telephone private branch exchange”, in Eighth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY’07), Los Alamitos, (2007), pp. 251-5.
- [6] A. Hammoud and D. Bourget, “Integrating Asterisk with InRule to Detect Suspicious Calls”, in Sixth Advanced International Conference on Telecommunications (AICT 2010), Los Alamitos, 2010, pp. 153-60.
- [7] K. S. Chava and J. Ilow, “Integration of open source and enterprise IP PBXs”, in 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities – TridentCom., Piscataway, (2007), pp. 70-5.
- [8] W. Xiaolong, “Comparison of open-source cloud management platforms: OpenStack and OpenNebula,” in 9th International Conference on Fuzzy Systems and Knowledge Discovery, 29-31 May 2012, Piscataway, (2012), pp. 2457-61.
- [9] R. Nasim and A. J. Kassler, "Deploying OpenStack: Virtual Infrastructure or Dedicated Hardware," in IEEE 38th International Computer Software and Applications Conference Workshops, Karlstad, (2014), pp. 84-89.

- [10] F. Wuhib, "Dynamic resource allocation with management objectives-Implementation for an OpenStack cloud", in 8th International Conference on Network and Service Management (CNSM 2012), Piscataway, (2012), pp. 309-15.
- [11] N. Ueno, "FreeCloud: A Trial Service for OpenStack (vol 9) [Online]", Available: https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201112fa6.pdf&mode=show_pdf, (2011).
- [12] O. Litvinski and A. Gherbi, "Openstack scheduler evaluation using design of experiment approach", in IEEE 16th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), Piscataway, (2013), pp. 7.
- [13] J. A. L. del Castillo, "OpenStack Federation in Experimentation Multi-cloud Testbeds", in IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), Los Alamitos, (2013), pp. 51-6.
- [14] D. Bernstein, "Cloud Foundry Aims to Become the OpenStack of PaaS", Cloud Computing, IEEE , vol.1, no.2, pp.57,60, July 2014 doi: 10.1109/MCC.2014.32
- [15] V. Liu, "An architecture for enhanced assurance in e-health systems", Ph. D. dissertation, Dep. Scie. Eng., Queensland Uni of Tech., Brisbane, QLD, (2011).
- [16] T. Rosado and J. Bernardino, "An overview of openstack architecture", in the 18th International Database Engineering & Applications Symposium, Porto, (2014), pp. 366-367.
- [17] Openstack, (2013, May). Openstack Architecture [Online], Available: http://docs.openstack.org/icehouse/install-guide/install/yum/content/ch_overview.html
- [18] Cisco, SUSE Cloud Integration with Cisco UCS and Cisco Nexus Platforms [Online]. Available: <http://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-c220-m3-rack-server/white-paper-c11-731115.html>
- [19] A. D. Tesfamicael, "Implementation and evaluation of open source unified communications for SMBs", In International Conference on Computational Intelligence and Communication Networks, Udaipur, (2014), pp. 1243 – 1248.

Authors



Aklilu Tesfamicael, he received the B.S. degree in mathematics from University of Asmara, Asmara, Eritrea, in 1997 and the M.S. degree in information technology from University of South Australia, Adelaide, Australia, in 2003. He is currently pursuing the Ph.D. degree in information technology (unified communications and collaboration security) at Queensland University of Technology (QUT). He is also cisco certified professional in voice, routing and switching. He has worked in IT Industry, large enterprise and government departments for over fifteen years, attaining the position of Network, Unified Communications and Systems Engineer, and Solutions Architect. His research interest is in Unified Communications Security on-premise and in the Cloud.



Vicky Liu, she is a Lecturer in the Science and Engineering Faculty at Queensland University of Technology, Australia. Her PhD dissertation proposed security architecture to facilitate the enforcement of privacy and security in such systems. Vicky has research interests in topics such as network security and performance metrics, security services for Unified Communications, network security for next-generation networks, and system architectures for health information systems. Vicky's teaching areas are Internet services and protocols, networking planning and next-generation networks. She supervises PhD/Masters Research/ Honours students as well as many project students. She is also a certified Cisco Certified Network Associate (CCNA) instructor.



William Caelli, he is an Adjunct Professor in the Science and Engineering Faculty of the Queensland University of Technology and is a Director of International Information Security Consultants Pty Ltd. He has more than 50 years of experience in the IT industry in business, education and research areas. He was the founder in 1979 of ERACOM Pty Ltd, an information security company, and of the Information Security Research Centre in 1988 at the then Queensland Institute of Technology, now the Queensland University of Technology. He is member of the Board of the Colloquium for Information Systems Security Education based in Maryland, USA. He also currently chairs the Stability and Security Advisory Committee (SSAC) of auDa, Australia's domain name administration authority for the Internet. He has been a member of IFIP's Technical Committee 11 (Information Security) since 1984, being its Chair for 6 years.