Implementation and Evaluation of Open Source Unified Communications for SMBs

Aklilu Daniel Tesfamicael, Vicky Liu, William Caelli, Jonathan Zureo
Science and Engineering Faculty
Queensland University of Technology
Brisbane, Australia
aki.tesfamicael@hdr.qut.edu.au, {vicky.liu, w.caelli}@qut.edu.au, jonathan.zureo@connect.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. In this paper, we will discuss and demonstrate an open source UC solution, viz. "Asterisk" for use by SMBs, and report on some performance tests using SIPp. The contribution from this research is the provision of technical advice to SMBs in deploying UC, which is manageable in terms of cost, ease of deployment and support.

Keywords*— Unified Communications (UC); Voice Over IP (VoIP); Asterisk; Open Source VoIP; Session Initiation Protocol (SIP)

I. 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 [5], are pushing their products to the market with associated high prices which have become too steep for the small and medium sized business (SMBs).

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. 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. 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". 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 SMBs 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 for SMBs. Section V gives a description of the experimental lab used followed by an analysis of results in Section VI. The paper closes with conclusions and proposals for future work, in Section VII.

II. 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 Asterisk such as voice messaging, conferencing and mobility. Gohel and Lakhtaria [4] 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 [8] 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 [1] 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 which can be incorporated with our proposed design solution for SMBs. In [6], Hammoud and Bourget 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 [2] reported on the successful integration of an Asterisk server with the Cisco Call Manager for interoperability.

III. REAL CASE SENARIOS

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. *Senario 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. Recently 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.

C. *Scenarios Analysis*

What we have learned from these two real case scenarios is that the deployment of UC 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 [5] are expensive. Therefore, this research proposes an alternative implementation for SMBs realizable on an open source platform.

IV. 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 [4]. 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 [1].

A. UC Overiew

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 for an SMB is illustrated in Fig.1.

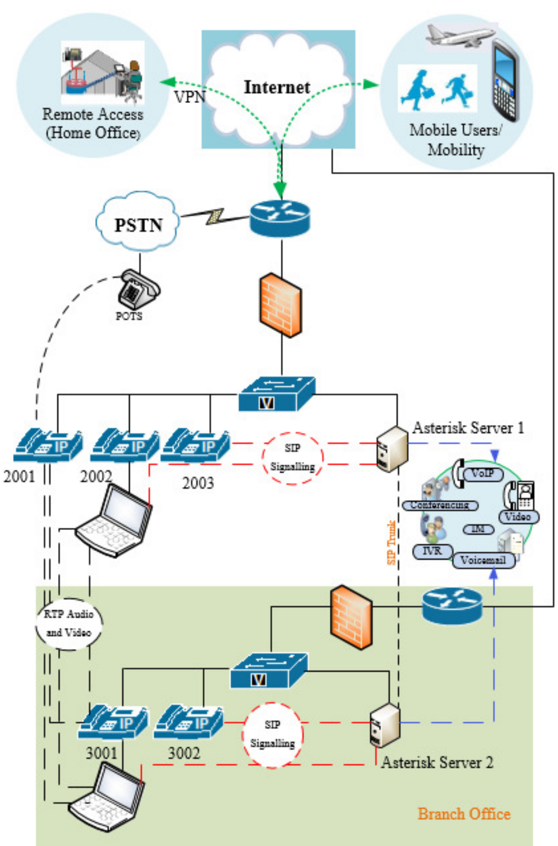

Fig. 1. High Level Design – Proposed Design Solution

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 SIPconnect to eliminate or minimize the need for a costly gateway.

B. 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 [3]. 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 are 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 Fig. 2.

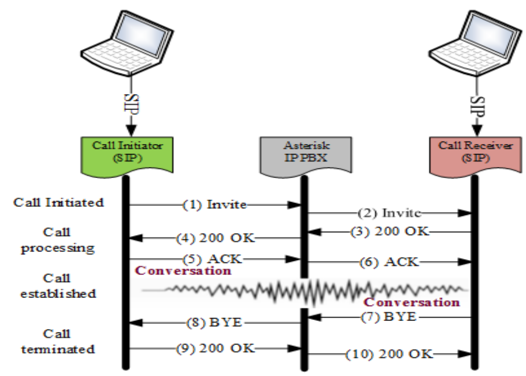

Fig. 2. SIP Session Signal Flow Diagram

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 [7].

In Asterisk, channels are created using Channel Drives and call processing is centered around those channel drivers [10] which handle all the protocol-specific details of ISDN, SIP, and other telephony protocols and communicate with Asterisk. Fig. 4 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.

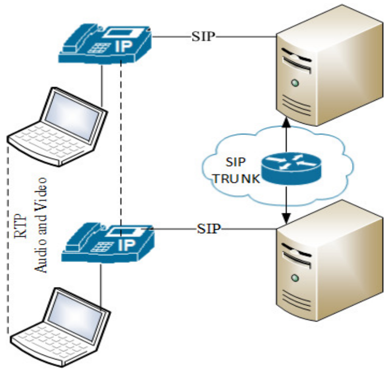

Fig. 4. RTP Audio and Video Flow Diagram

V. 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.

Fig. 5 shows the high level design of the test lab we implemented. The hardware specifications are listed in Table I.

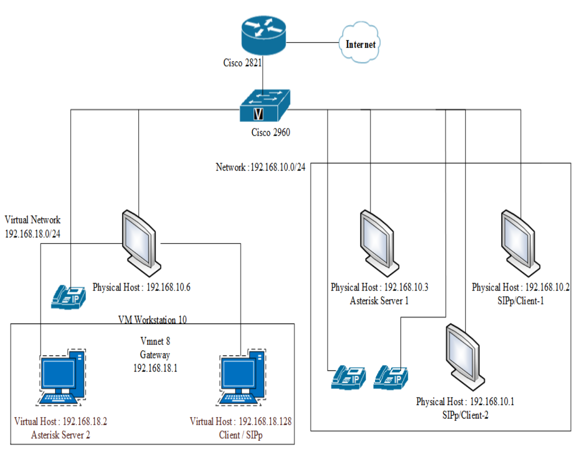

Fig. 5. Test Bed High Level Design

TABLE I. TEST LAB DESIGN HARDWARE SPECIFICATIONS

3 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:** 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

VI. 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 [10] and for a client we installed "Spark", an Open Source cross-platform IM client.

A. Voice and Voice Call Test

Fig. 6 shows a phone call conversation between two IP soft-phones based on the G.711 standard.

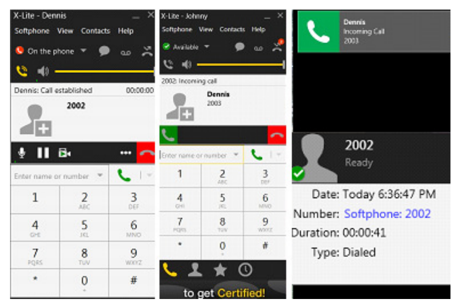

Fig. 6. Voice to Voice Call using G.711 Codec

Fig. 7 shows a video call conversation between two IP softphones based on the G.711 standard.

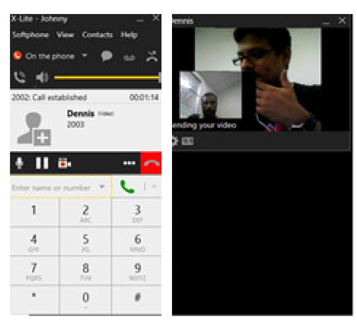

Fig. 7. Video to Video Call using G.711 Codec

B. *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 [11]. 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 II shows the configuration of "sip.conf" and "extension.conf" after we made the changes.

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

sipp –sn uac –d 20000 –s 3001 192.168.10.105 –l 30

This command connects as a client, gives the duration of the call 20 seconds, dials the IP address of the Asterisk server, 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. Fig. 8 summarizes the comparison of results revealed using G. 711 and Transcoding.

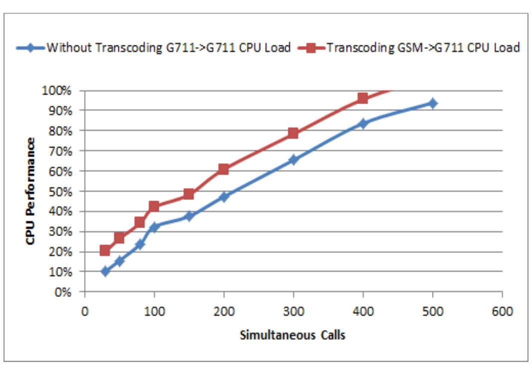

Fig. 8. Intel Core i7 3.40GHz CPU Performance on G711 and Transcoding

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 Fig. 9.

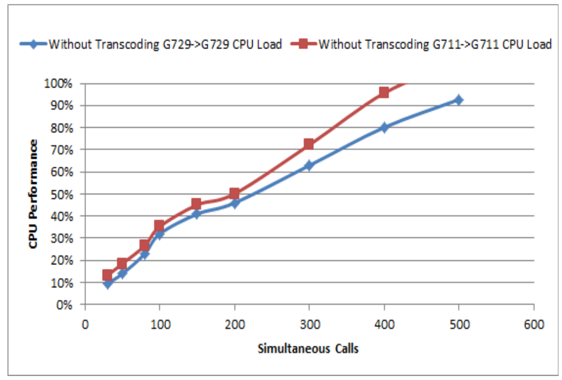

Fig. 9. Intel Core i7 3.40GHz CPU Performance on G729 and G711

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.

VII. CONCLUSION AND FUTURE WORK

The implementation guidelines we are proposing for SMBs is manageable in terms of cost, ease of deployment and support. Anyone with fundamental data networking skills can get the basic Asterisk System up and running within a short period of time. However, dial-plan configuration and full support for the system require advanced knowledge of that system. We recommend SMBs involve a UC professional to provide technical system support or to engage the vendor or UC service provider under a support level agreement.

In future work, we plan to implement Asterisk in a cloud environment and to investigate enhance security architecture for UC positioning to a cloud environment

AKNOWLEDGMENT

Our UC implementation and performance testing has been conducted on a networking laboratory with student-formed teams. The authors appreciate the contributions from Jiexuan Chen, Fabien Chastel and Dennis Pramod.

REFERENCES

[1] 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), 13-15 June 2007*, Los Alamitos, CA, USA, 2007, pp. 251-5.

[2] K. S. Chava and J. Ilow, "Integration of open source and enterprise IP PBXs," in *2007 3rd International Conference on Testbeds and Research Infrastructues for the Development of Networks and Communities - TridentCom '07, 21-23 May 2007*, Piscataway, NJ, USA, 2007, pp. 70-5.

[3] SIP [Online]. Available: http://www.voip-info.org/wiki/view/SIP/.

[4] 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, pp. 1-11, 01/ 2010.

[5] Magic Quadrant for Unified Communications [Online]. Available: http://www.ucstrategies.com/unified-communications-strategies-views/gartner-2013-uc-magic-quadrant-maturity-challenges-directions-for-the-future.aspx/.

[6] A. Hammoud and D. Bourget, "Integrating Asterisk with InRule to Detect Suspicious Calls," in *Sixth Advanced International Conference on Telecommunications (AICT 2010), 9-15 May 2010*, Los Alamitos, CA, USA, 2010, pp. 153-60.

[7] FreePBX [Online]. Available: www.voip-info.org/wiki/view/freePBX/.

[8] 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.

[9] Asterisk Architecture, The Big Picture [Online]. Available: https://wiki.asterisk.org/wiki/display/AST/Asterisk+Architecture,+The+Big+Picture/.

[10] OpenFire [Online]. Available: http://www.igniterealtime.org/projects/openfire/

[11] SIPp [Online]. Available: http://sipp.sourceforge.net/.