NTNU – Trondheim
Norwegian University of
Science and Technology

Scalability Modeling for Optimal Provisioning of Data Centers in Telenor

A better balance between under- and
over-provisioning

Knut Helge Rygg

Master of Science in Computer Science
Submission date: July 2012
Supervisor: Gunnar Brataas, IDI
Co-supervisor: Geir Millstein, Telenor GID
Terje Molle, Telenor Norge

Norwegian University of Science and Technology
Department of Computer and Information Science

Scalability Modelling for Optimal Provisioning of Data Centers in Telenor

Knut Helge Rygg

Supervisors:
Gunnar Brataas, SINTEF/NTNU
Geir Millstein, Telenor GID
Terje Molle, Telenor GID

July 18, 2012

Acknowledgements

This thesis would not have been possible without the help from several helpful persons.

I would like to thank my main supervisor, Adjunct Associate Professor Gunnar Brataas. Gunnar has given me many useful advices in the planning and execution of this thesis. He has pointed me in the right direction while still giving me the freedom to walk my own steps. Gunnar has great insight and experience from performance engineering combined with a genuine academic interest. His close follow-up during the whole project period has given me the motivation to carry out the task.

Terje Molle and Geir Millstein from Telenor Group Industrial Development has done an excellent job as co-supervisors. Terje and Geir have helped me refining and specifying the task so that the research questions has great value for both Telenor as a company and for the broad public as an academic contribution. Without Terje and Geir I would not have gotten the computer resources used to carry out this thesis.

I would also like to express my gratitude to Georg Lothe and Alexander Botnen, system administrator at Telenor. Georg and Alexander have had the main responsibility of granting me access and control of the computer test environment. They have installed new components and adjusted system configuration to fit my demands. You have always had a solution to the many small problems that arose during the setup and measurement phase. Without your help, there wouldn't be a single performance result.

Thank you all for your contributions

Abstract

The scalability of an information system describes the relationship between system capacity and system size. This report studies the scalability of Microsoft Lync Server 2010 in order to provide guidelines for provisioning hardware resources. Optimal provisioning is required to reduce both deployment and operational costs, while keeping an acceptable service quality.

All Lync servers in the test setup are virtualized using VMware ESXi 5.0 and the system runs on a Cisco Unified Computing System (UCS) platform. The scenario is a typical hosted Lync deployment with external users only and telephone integration. While several companies offer hosted virtual Lync deployments and the Cisco UCS platform has a rich market share, Microsoft's capacity planning guides don't provide help for such a deployment scenario or hardware platform. This report consequently fill an information gap.

The scalability is determined by using empirical measurements with different workloads and system sizes. The workload is determined by the number of Lync end-users and the system size varies from 1/4 to 4/4 Cisco UCS blade server. The results show a linear scaling in the range of 1/4 to 4/4 blade servers. The processor is found to be the main bottleneck resource in this deployment. The mean opinion score (MOS) as well as the front end server utilization are the best metrics for monitoring service quality.

Summary in Norwegian

Skalerbarheten i et informasjonssystem beskrives som forholdet mellom systemets kapasitet og størrelse. Denne rapporten studerer skalerbarheten i Microsoft Lync Server 2010 med det formål å gi retningslinjer for dimensjonering av maskinvareressurser. Optimal dimensjonering er nødvendig for å redusere både implementasjons- og driftskostnader, samtidig som en akseptabel tjenestekvalitet opprettholdes.

Alle Lync-servere i testoppsettet er virtualiserte med VMware ESXi 5.0 og systemet kjører på en plattform bygget opp av Cisco Unified Computing System (UCS). Scenariet tilsvarer en typisk Lync implementasjon som settes opp som en kundetjeneste. Det betyr at det bare er eksterne brukere i systemet og telefon-integrasjon følger med. Det finnes flere selskaper som tilbyr Lync som en kundetjeneste og Cisco UCS-plattformen har en stor markedsandel. Likevel gir ikke Microsoft sitt kapasitets-planlegging-verktøy noen hjelp for å dimensjonere en slik implemtasjon. Denne rapporten fyller dermed et informasjonsbehov.

Skalerbarheten til Lync bestemmes ved hjelp av empiriske målinger med ulike belastninger og systemstørrelser. Belastningen er bestemt av antall Lync-brukere og systemstørrelse varierer fra 1/4 til 4/4 Cisco UCS bladservere. Resultatene viser at det er en lineær skalering i området 1/4 til 4/4 bladservere. Prosessoren er den største flaskehalsen i denne implementasjonen. Mean opinion score (MOS) samt prosessorutnyttelsen på front end server er de beste parameterne for overvåking av tjenestekvalitet i Lync.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This master thesis is a contribution to an ongoing research collaboration project between Telenor and SINTEF that was started in 2010. One of the goals in this project is to reduce the overall power consumption in Telenor's data centers. The preliminary findings in this research project is described in a Telenor internal report [10]. The report describes how correct provisioning in server deployment can help reduce both the operational costs related to power consumption as well as hardware investments costs. Correct provisioning can thereby help reduce the total life cycle cost of an information system. Optimal provisioning is a non-trivial task that may require a lot of research for each information system under concern.

1.1 Task description

The following problem formulation is based on this information requirement and provides the starting point for this thesis:

> *Telenor has several data centers which delivers important services to its customers. The prime objective of this task is to avoid under provisioning (too little hardware resources to handle the workload, so that response times gets too long) and at the same time to reduce over provisioning (too much hardware resources to handle the workload, so that utilization is low and the equipment including energy usage gets costly). It is first important to find out how this can be solved using all the data Telenor possess, and next to make a scalability model which can assist in achieving the prime objective. This model has to make some crude approximations. If possible, the predictions models shall be compared with actual system behavior (validation). An important result will be a method supporting the prime objective. This method shall also describe the amount of manual work required to perform this task. This task is ambitious, therefore achieving only parts will make a good contribution, as long as the missing parts are sketched.*

The problem description describes a goal of better provisioning in data centers without mentioning which of Telenor's information systems that should be studied. To limit the work in this master thesis a specific information system had to be chosen. Microsoft Lync Server 2010 was a good candidate because (1) Telenor wanted more information on its behaviour in order to do optimal provisioning and (2) a scalability study on the specific fully virtualized instance would have scientific value to the broad public.

1.2 Research questions

Based on the problem description the following resarch questions were formulated in accordance with the supervisors:

1. *How does a fully virtualized*[1] *instance of Lync scale when run on the Cisco UCS platform*[2]*?*
2. *What is the bottleneck resource* [3] *for this Lync implementation?*
3. *What metrics*[4] *are best suited to monitor service quality?*

The research questions are further described and discussed in the following subsections.

Scaling

Telenor hosts Lync deployments for several customers. All servers in the Lync deployment are virtualized because the virtualization technology provides great flexibility in terms of hardware usage as well as less maintenance work. A virtualized deployment on a custom hardware platform cannot benefit directly from the server requirements defined by Microsoft. Therefore, creating a scalability model will help Telenor provisioning the right amount of hardware resources based on customer needs. Scalability defines the relationship between system capacity and system size. This may be measured by number of operations or number of simultaneous end-users. The latter is the most reasonable metric for Lync. The system size describes how much resources the information system possesses to process the load while maintaining an acceptable quality of service. While a batch system may run on full utilization to achieve the best performance, an interactive system like Lync will have reduced quality of service if the utilization is too high.

If the capacity of an information system is proportional to the system size, the system is called linearly scalable. The scalability is sub-linear if the system capacity grows at a slower rate than system size. This is the most common case of scalability. A system may also be linearly scalable only within a given system size range. Finding out at which system size ranges the scaling is linear and when it is sub-linear provides useful information for host providers. Serving Lync instances in the linear scaling range is

[1] A fully virtualized instance of Lync, means that all server roles are implemented as virtual machines. There are no Lync services running on physical servers.

[2] The Cisco UCS platform is the hardware platform used for the tests in this master thesis. Another platform with a different hardware architecture may give different results regarding scalability and provisioning.

[3] The resource with the highest utilization

[4] System administrator needs relieable metrics to monitor service quality in order to keep business critical services available.

more cost-effective than in a sub-linear scaling range. Alternatively, the system size would require a different hosting pricing.

The scalability of Lync can be found by either creating a scalability model or by running empirical performance measures. This thesis aims to create a scalability model and validate the model using empirical testing. The scalability model will help Telenor to provision resources when setting up a new Lync instance for a customer. Having correct knowledge of provisioning will enable the best utilization of available hardware resources without affecting service quality.

Bottleneck resource

The computational resources of an information system can be divided into several categories; processing, memory, network and storage. Some systems may require a lot of network traffic to be exchanged while others are more dependant on memory or processing. It is always a goal to provision the right amount of resources of each category so that each resource group is equally utilized. The resource group with the highest utilization will become a bottleneck and increasing the resources of the bottleneck is the only way to improve performance. To give an example, upgrading from 1 Gbit/s to 10 Gbit/s network interfaces will not increase performance if the processing resources are the bottleneck.

Better knowledge of resource demands on each of the different resource types can help design more optimal hardware modules. The Cisco UCS platform is customizable, so that extra CPU, memory, network and storage resources can be added as required. Identifying bottleneck resources can help Telenor upgrading the system by increasing a single resource instead of a wider upgrade of all resources. This will reduce the cost of upgrading or give a larger upgrade at the same price.

Service quality

A scalability model can help Telenor provision the right amount of resources to fit customer needs. However, the customer's use of the system can be highly variable. A customer may exceed their stated needs for a shorter period of time and thereby need more resources. On the other hand it is reasonable to believe that resource usage will decrease at night time, in weekends and in public holidays. In such conditions, virtualization technology proves useful by letting system administrators collecting several virtual machines on the same physical server and thereby freeing one or more physical servers that can be powered down to reduce costs.

To avoid removing to much resources, system administrators need reliable performance metrics that give a good indication of the actual service quality the customer experiences. A good service quality metric should be able to give an early warning when resource usage closes to maximum. An early warning can give time to mitigate the consequences before they happen by provisioning more resources.

1.3 Thesis layout

The following paragraphs outlines the remaining chapters of this report.

The background chapter starts of with an introduction to Microsoft Lync Server 2010. Then it discusses different approaches to finding the scalability of Lync and some of the challenges that must be solved. Other reports on Lync scalability or performance is reviewed as well as vendor performance guidelines.

The method chapter describes the specifications of the fully virtualized, hosted Lync instance that is tested in this report. Differences between the test instance and an actual instance is described and its impact on validity is discussed. This instance is also compared to instances built on different scenarios and the qualitative differences are sketched. The chapter describes two approaches to determine scalability. The first is an attempt to model the Lync instance using the Structure and Performance (SP) framework and the second is an empirical approach to determine system capacity versus system size based on actual measurements. It turns out that the SP framework can be used to design a model for the system, but it is hard to parametrize and measure. The details on how to determine system capacity, how to measure system performance, how to scale the system size, etc. is thoroughly described to allow reproduction of the test results.

The results chapter describes the results from the empirical measures. The impact of systematic or random errors in the actual measurements is discussed and related to the scalability findings. This section also contains an evaluation of the different quality metrics and gives guidelines on which metrics are most useful for system administrators that monitor system performance.

The discussion chapter reviews the results from the empirical tests and tries to compare them with results from a scalability model. It further discusses scaling beyond the system range that was measured in this thesis. A discussion of the time needed to reproduce the results is also included to help those trying to reproduce the results.

The conclusion chapter summarizes the important findings in this report and concludes on the research question. Some of the findings lead to new research questions that arise from the work in this report. These are outlined in further work and may give inspiration for another master thesis or other scientific work.

Chapter 2

Background

This chapter gives the reader an introduction to Microsoft Lync Server 2010. A good understanding of the system design along with information on different server roles and workload types is crucial to understand and evaluate the performance models and test setup. The chapter continues with a short literature study on Lync performance modelling to give shed light on what alternative scalability and performance evaluations exists.

2.1 Introduction to Microsoft Lync Server 2010

Microsoft Lync Server 2010 is a unifed communication system developed by Microsoft. Unified communication is the integration of both real-time and non-real-time communication services such as instant messaging, telephony, audio/video conferencing, data sharing, e-mail, etc. into a consistent unified user interface. There are several competing products in the open marked, such as Cisco UC, Avaya Aura, Alcatel-Lucent OpenTouch Communication Suite and Siemens OpenScape UC that offer similar services to Microsoft Lync. Gartner Group's Magic Quadrant for Unified Communications [4] rates Microsoft Lync as the best current UC product regarding completeness of vision and ability to execute. According to Gartner Group there are several strengths of Microsoft Lync: (1) A large number of Lync deployments, in which some of them have eliminated the need for a private branch exchange (PBX). (2) Microsoft's pending Skype acquisition offer, along with the Lync Online and the Office365 cloud service offerings, suggest that Lync will mature as a comprehensive and hybrid UC product. (3) Companies report that, once deployed, Lync functions can be integrated into business processes and applications, providing new, different and effective ways to perform tasks. In some cases, these new functions are achieved by deploying Lync enhancements from a growing list of ecosystem partners. (4) Microsoft has positioned Lync to compete in telephony markets by adding several partner telephone handsets, by bundling of basic Lync functions in the Core client access license (CAL) and by offering a specific Lync Voice CAL.

Microsoft Lync is a modular system that can take a number of different configuration depending on the level of functionality and capacity required. The following sections describe the topology and different server roles in Lync and what functionality they provide. The documentation of Microsoft Lync can be found on Microsoft's web pages [13].

2.1.1 Topology

The Microsoft Lync architecture is composed of a central site server park that connects to one or several branch sites. Figure 2.1 shows this topology. Clients can either be connected to the same local area network as the servers or they can connect from Internet through the edge server. A PSTN gateway allows users to make outbound calls to the public switched telephone network (PSTN) as well as allowing inbound PSTN calls to the UC network.

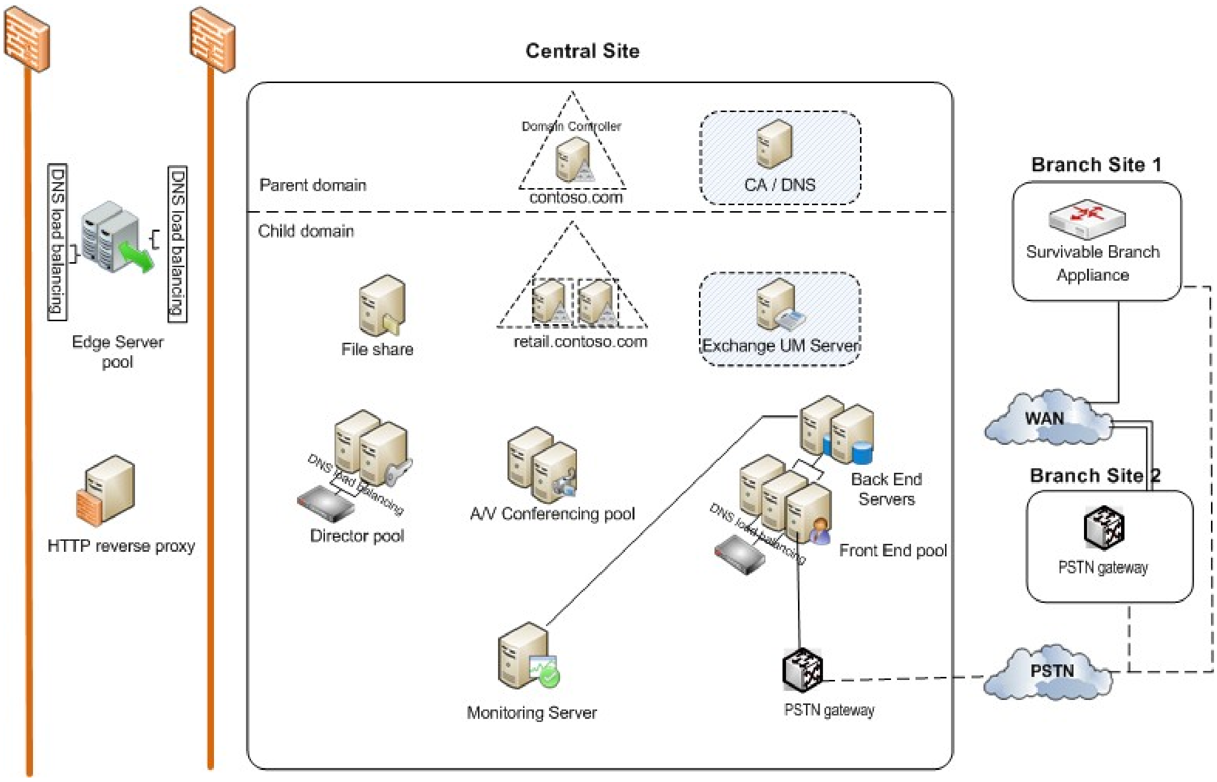

Figure 2.1: Reference topology with high availability and a single data center [7]

2.1.2 Server roles

A Lync deployment can contain several server roles. The server roles can be deployed separately on different servers or some roles may be collocated to reduce hardware usage. For increased performance some server roles may be deployed in server pools. A server pool uses load balancing to share the load among several servers. Figure 2.1 shows many of the server roles, while some server roles (mediation server and archiving server) are collocated with the front end server pool.

Front end server

The front end server role is the core server role and runs many basic Lync Server functions. Front end server includes the following functionality:

- User authentication and registration
- Presence information and contact card exchange
- Address book services and distribution list expansion
- IM functionality, including multiparty IM conferences
- Web conferencing and application sharing (if deployed)
- Application hosting services, for both applications included with Lync Server (for example, Conferencing Attendant and Response Group application) and third-party applications

Additionally, one front end pool in the deployment also runs the central management server, which manages and deploys basic configuration data to all servers running Lync server 2010. The central management server also provides Lync server management shell and file transfer capabilities.

Back end server

The back end servers are database servers running Microsoft SQL server that provide the database services for the Front End pool. A single back end server can be deployed, but a cluster of two or more servers is recommended for failover. Back end servers do not run any Lync server software.

A/V conferencing server

The A/V conferencing server provides A/V conferencing functionality. A/V conferencing role can be collocated with front end server or deployed separately as a single server or A/V conferencing server pool.

Edge server

Edge server enables your users to communicate and collaborate with users outside the organization's firewalls. These external users can include the organization's own users who are currently working off site, users from federated partner organizations, and outside users who have been invited to join conferences hosted on your Lync Server deployment. Edge Server also enables connectivity to public IM connectivity services, including Windows Live, AOL, and Yahoo. The Edge server can be deployed with three external network interfaces. One interface for SIP traffic (IM, presence, signalling, etc), another for web conferencing and a third for A/V functionality.

Mediation server

Mediation server is a necessary component for implementing enterprise voice and dial-in conferencing. Mediation server translates signaling and, in some configurations, media between the internal Lync server infrastructure and a public switched telephone network (PSTN) gateway, IP-PBX, or a Session Initiation Protocol (SIP) trunk. The mediation server role can be collocated with the front end server.

Monitoring server

Monitoring server collects data about the quality of the network media, in both enterprise voice calls and A/V conferences. This information can help provide the best possible media experience for users. It also collects call error records (CERs), which can be used to troubleshoot failed calls. Additionally, it collects usage information in the form of call detail records (CDRs) about various Lync Server features. CDRs are collected from both the front end server as well as the mediation server and sent automatically to the monitoring server destination queue.

Archiving server

Archiving server enables archiving of IM communications and meeting content for compliance reasons. This server role is not needed unless legal compliance concerns require the company to do so. The archiving server role can be collocated with the front end server.

Director

Directors can authenticate Lync Server user requests, but do not home user accounts, or provide presence or conferencing services. Directors are most useful in deployments that enable external user access, where the Director can authenticate requests before sending them on to internal servers. Directors can also improve performance in organizations with multiple Front End pools. The director is not a required component.

HTTP reverse proxy

The HTTP reverse proxy server role is a required component to enable external users to download content from the internal Lync servers. The reverse proxy can be installed using Microsoft Threat Management Gateway. The reverse proxy provides the following functionality:

- Enabling external users to download meeting content for meetings.
- Enabling external users to expand distribution groups.
- Enabling remote users to download files from the Address Book service.

- Accessing the Microsoft Lync Web App client.
- Accessing the Dial-in Conferencing Settings webpage.
- Accessing the Location Information service.
- Enabling external devices to connect to Device Update web service and obtain updates.
- Enabling mobile applications to automatically discover mobility URLs from the Internet.

Domain controller

All Lync servers (except the reverse proxy server) are members of the same Windows active directory domain. A domain controller is used to manage servers and users. The domain controller server also contain the Kerberos Key Distribution Center (KDC) and Kerberos Authentication Server (AS) which is required for authenticating communication among servers.

2.1.3 Hosted Lync

While some larger companies choose to manage their own Lync server instance there is also a marked for *hosted Lync*. Hosting companies can provide Lync *as a service* so that customers get all the functionality without worrying for hardware or software. Customers of hosted Lync only need to install the client software on PCs, MACs and/or IP-telephones and connect via Internet or a dedicated WAN connection. A typical hosted Lync instance consists of a single central site and all Lync users are external.

2.2 Capacity planning for Microsoft Lync

There are two different categories of performance evaluation descriptions available for Microsoft Lync. (1) Vendor recommendations developed by Microsoft and (2) scientific papers based on measurements of Lync instances. The first category includes a capacity planning guide and a scenario based capacity calculator. The second category includes a network performance report on Microsoft Office Communication Server (OCS), the predecessor of Microsoft Lync and a sizing study of a virtualized Lync instance.

2.2.1 Vendor recommendations

Capacity planning guide

Microsoft has written a capacity planning guide [13] to help system administrators provision the right amount of hardware resources for their Lync instances. The main

Table 2.1: Standard configuration server hardware requirements [14]

General Lync server 2010 server roles	
Processor	64-bit dual processsor, quad-core 2.0 GHz or higher or 64-bit 4-way processor, dual-core, 2.0 GHz or higher. Intel Itanium processors are not supported for Lync Server 2010 server roles.
Memory	16 GB of RAM
Storage	Local storage with at least 72 GB free disk space on a 10,000 RPM (rotations per minute) disk drive
Network	1 network adapter required (2 recommended), each 1 Gbps or higher
Director server role	
Processor	64-bit dual processsor, quad-core 2.0 GHz or higher or 64-bit 4-way processor, dual-core, 2.0 GHz or higher. Intel Itanium processors are not supported for Lync Server 2010 server roles.
Memory	4 GB of RAM
Storage	Local storage with at least 72 GB free disk space on a 10,000 RPM (rotations per minute) disk drive
Network	1 network adapter required (2 recommended), each 1 Gbps or higher
Back end servers and other database servers	
Processor	64-bit dual processsor, quad-core 2.0 GHz or higher or 64-bit 4-way processor, dual-core, 2.0 GHz or higher
Memory	32 GB recommended for Back End Server (with or without collocated Archiving and Monitoring databases), 16 GB recommended for Archiving and Monitoring database (not collocated with the Back End Server).
Storage	Local storage with at least 72 GB free disk space on a 10,000 RPM (rotations per minute) disk drive
Network	1 network adapter required (2 recommended), each 1 Gbps or higher

parameter to provision hardware for each server is the number of Lync users. The planning guide is based on a typical user model [2] so that the average load generated by each user is known. The user model is developed to reflect a typical organization using Lync, but it is natural to assume that there can be deviations from organization to organization. To exemplify some of the parameters, the user model assumes that 80% of the users are logged in at the same time. 70% of the users are internal users while the remaining 30% are connected externally. It also specifies the number of calls per day, number of instant messaging conversations, presence updates etc. The recommendations in the planning guide are only suitable if the user model is representative for the actual workload.

The recommendations in the capacity planning guide tells the system administrator how many users are supported if a server role is deployed on one or more standard configuration servers. The number of servers can then be adapted to fit the expected load. The standard configuration hardware platform reflects a typical off-the-shelf server with the specifications given in table 2.1.

The recommendations are based on physical server installation. If the server roles are virtualized, the number of users supported are roughly halved. This recommendation may seem strict because virtualization doesn't necessarily degrade performance by a rate of two. The natural explanation of the halving effect is that the Microsoft Hyper-V hypervisor doesn't support allocating more than 4 virtual CPUs (vCPU) to each virtual machine, while the standard configuration hardware platform uses 8 cores. Thereby there is roughly a halving in processor performance. The recommendations doesn't concern that several virtual machines can be run on the same host (e.g. several front end servers in a front end pool [19]). VMware, a competing hypervisor, is not limited to 4 vCPU per virtual machine but supports up to 32. Thereby it can use all 8 cores in the standard configuration server and the performance recommendations for virtual deployments are invalidated. Virtualization comes with a certain overhead, but the degradation should be lower than two.

Telenor uses a full virtualization of all server roles. In addition several servers are allowed to run on the same physical server. The hardware platform is based on Cisco Unified Computing System where the hardware specification of the server blades differ significantly from Microsofts standard configuration server. The recommendations in the capacity planning guide is therefore of little use.

Capacity planning calculator

Microsoft has also published a scenario based capacity planning calculator [15] that allows more fine-grained analysis of Lync capacity requirements. This is a spreadsheet that provides granular capacity numbers for each workload in a variety of situations. The system designer can specify both the number of users and a custom user model by varying a number of different parameters. The output is a recommendation on the number of servers to use and their respective CPU utilization, memory usage, etc. The calculations are still based on the same standard configuration hardware described in the capacity planning guide and the percentage of external users is hard coded to 30%. This does not fit with Telenor's hosted scenario where 100% of the users are external. The resource demand on the edge server will have a higher resource demand than the calculator falsely indicates. Even though the capacity calculator allows more flexibility, it is not sufficient to provide good estimates for Telenor's deployments.

Provisioning of Microsoft Lync is described in books explaining Microsoft Lync deployment [11, 23]. However, they only reference Microsoft's own capacity planning guidelines. Thereby there is little help for those planning to use custom hardware or specialized scenarios.

2.2.2 Scientific performance studies on Lync scenarios

Microsoft Lync was released in Nov, 2010 and few scientific papers that describe performance evaluation of the system are yet published. There are however two papers worth noting.

The first report was released in 2011 and describes the bandwith usage in Microsoft Office Communication Server (OCS) which is the predecessor of Microsoft Lync. Microsoft OCS has much of the same functionality as Microsoft Lync, except the telephone integration introduced Lync. In his report [18], Patrick Selling studies the bandwith usage in a Microsoft (OCS) instance at an academic institution in the upper Midwest. The main finding in this paper is that the overall server load in terms of network traffic can be estimated from the connected user endpoints and their respective traffic patterns. The paper shows that the peak network utilization can be quite different from the average so using mean values may be inaccurate for bandwidth provisioning. This paper supports Microsofts recommendation that implies that network usage is proportional to the number of users. However the report gives little information on provisioning other hardware resources like CPU, memory or storage. Thereby, the report has little value for overall provisioning.

The second paper [19] is called a sizing study of Microsoft Lync. This is a report that is conducted by Global Solutions Engineering on behalf of DELL. The aim of the report is to make the most out of Dell hardware running Microsoft Lync Server. The paper describes a test environment built by Dell PowerEdge R720 servers. Back end servers are installed on two physical PowerEdge R720 servers for redundancy. The front end servers are with collocated A/V server and mediation server role and deployed as virtual machines on two PowerEdge R720 servers. Archiving and monitoring server roles where deployed on a separate server. All VM's are stored on an a SAN as well as the back end database and archiving/monitoring database. User load is simulated using Microsoft Lync Stress and Performance Tool which is discussed in further detail in the next chapter. The server hardware configuration of the deployment is shown in table 2.2 and a network architecture is shown in figure 2.2

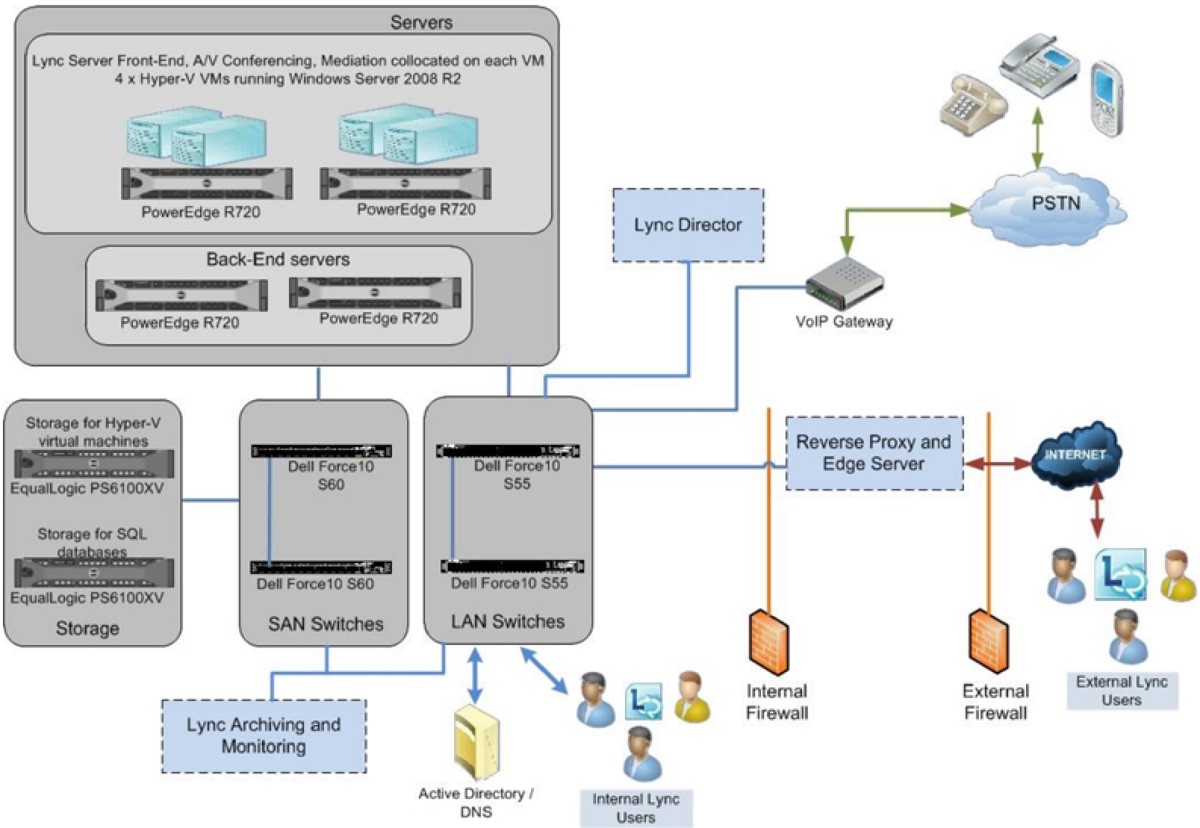

Figure 2.2: Reference architecture for Dell sizing study of Lync [19]

Table 2.2: Reference configuration for Dell sizing study of Lync [19]

Server configuration	**Detail**
Microsoft Lync Server Version	Enterprise Edition
Physical Server Configuration (Host)	2 x R720 2 x 8-core Intel Xeon 24 x 4 GB = 96 GB Memory 2 x 146GB 15k SAS
VM Configuration: Front End, Mediation and A/V Conferencing Server Roles	4 x Hyper-V Windows Server 2008 R2 VM 2 x VMs per host 4 vCPUs per VM 16 GB Memory per VM
Back-End Server	2 x R720 (in a fail-over cluster) 2 x 6-core Intel Xeon 16 x 2 GB Memory = 32 GB
Storage Configuration	**Detail**
Storage for Hyper-V VM?s	Dell EqualLogic PS 6100XV iSCSI SAN 24 x 146GB 2.5″ NL-SAS in RAID 10
Storage for Back-End Database, Archiving/Monitoring Database	Dell EqualLogic PS 6100XV iSCSI SAN 24 x 146GB 2.5″ NL-SAS in RAID 10
Additional Hardware	5 x Quad Port Network Interface Cards
Networking Configuration	**Detail**
LAN Networking	2 x Dell Force10 S55 Switches
SAN Networking	2 x Dell Force10 S60 Switches
VoIP Connectivity	PSTN Gateway or SIP Trunking
Optional Components	**Detail**
Additional Server Roles	Lync Server Director Pool Lync Server Archiving and Monitoring

The number of front end server virtual machines are varied from 1, 2 and 4 to increase the system size. For each configuration the maximum number of users are found. The report finds that the number of users scales linearly to the system size in the range of 3000 to 12000 users. The test results are shown in figure 2.3. The number of active connections is close to 3300, for each virtual machine, not 3000 like the number of users. This is not commented in the report, but probably comes from the multiple point of connection (MPOP) parameter that is default set to 10%. The MPOP parameter indicates that some of the users are logged in from multiple endpoints at the same time, e.g. from two different PCs.

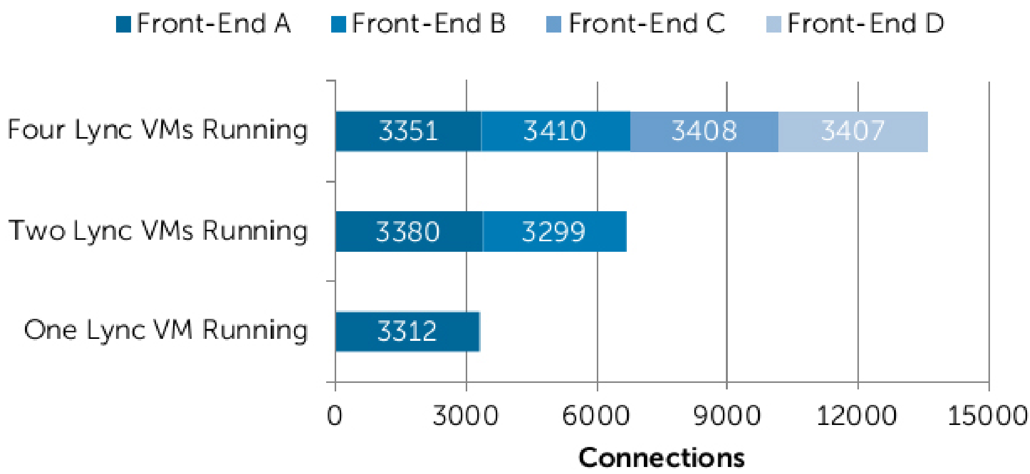

Figure 2.3: Number of connections versus system size [19]

Even though this report studies Lync scalability and gives actual recommendation on hardware provisioning, there are several differences when comparing to the Telenor deployment studied in this report:

1. **Licensing:** Dell's sizing study uses the enterprise edition of Lync which allows several servers to be assigned to the same pool. A total of four front end servers are allocated to the same front end pool increasing the number of users that can be served. This type of scaling is called scale out because several identical servers are added. This report uses the standard edition of Lync which does not support pools. Because of this, scaling out is not possible. The system size is varied by using another scaling method; scaling up. By scaling up, the resources of a server are increased to allow more users to be served.

2. **Server configuration:** Dell's sizing study uses dedicated servers for the back end server role. The A/V and mediation server roles are collocated with the front end server. This report uses a front end server that collocates the back end server role. This is a requirement in the standard edition license. The A/V server role is also collocated with the front end server, while the mediation server is deployed on a separate server. This report also includes the edge server and reverse proxy server role which is excluded in Dell's sizing study.

3. **Virtualization:** Dell's sizing study uses virtual machines for the front end servers while the remaining servers are installed on physical machines. This report studies a fully virtualized environment where all Lync servers and databases are vir-

tualized. A fully virtualized environment benefits from ease of management, increased redundancy and possibly better hardware utilization. The drawback is however that the virtualization add an overhead layer that may reduce performance.

4. **User scenario:** Dell's sizing study has 100% internal users. Internal users don't need to use edge or reverse proxy server so the performance of these servers can be neglected. This report which is based on a hosted Lync scenario uses 100% external users and therefore represents a different load on edge and reverse proxy server.

Chapter 3

Method

3.1 Test environment

This section describes the configuration of the test environment that was used to measure scalability and find provisioning guidelines. System configuration, hardware platform and virtualization framework are divided into separate subsections for better readability.

3.1.1 System configuraton

This Microsoft Lync deployment is specialised for hosted Lync. All servers are deployed at a single central site with a PSTN gateway. Users connect using an external connection to the edge server. A total of 7 servers were used. These are shown in table 3.1.

Table 3.1: Servers used in hosted Lync instance

Server name	**Description**
Front end server (standard edition)	This server has collocated the back end and A/V server role
Edge server	Allows external users to connect to internal servers
Reverse proxy	Allows external users to download content
Mediation server	Enables telephone integration. Assosiated with a PSTN gateway
Monitoring server	Collects CERs and CDRs to monitor QoE
SQL server	A database server for the monitoring server
Domain controller	Active directory domain controller and Kerberos KDC/AS

The central site is connected to the customer site using an IPT connection which is a dedicated WAN connection hosted by Telenor. An IPT connection is also used to connect the central site to the PSTN gateway. Firewalls are used to separate the internal Lync servers from the remaining network. External users can only access the edge server or reverse proxy server. This configuration increases security by protecting the internal Lync servers from network attacks. An overview of the system is shown in figure 3.1. The green arrows are added to indicate which servers or subnetworks that are allowed to communicate.

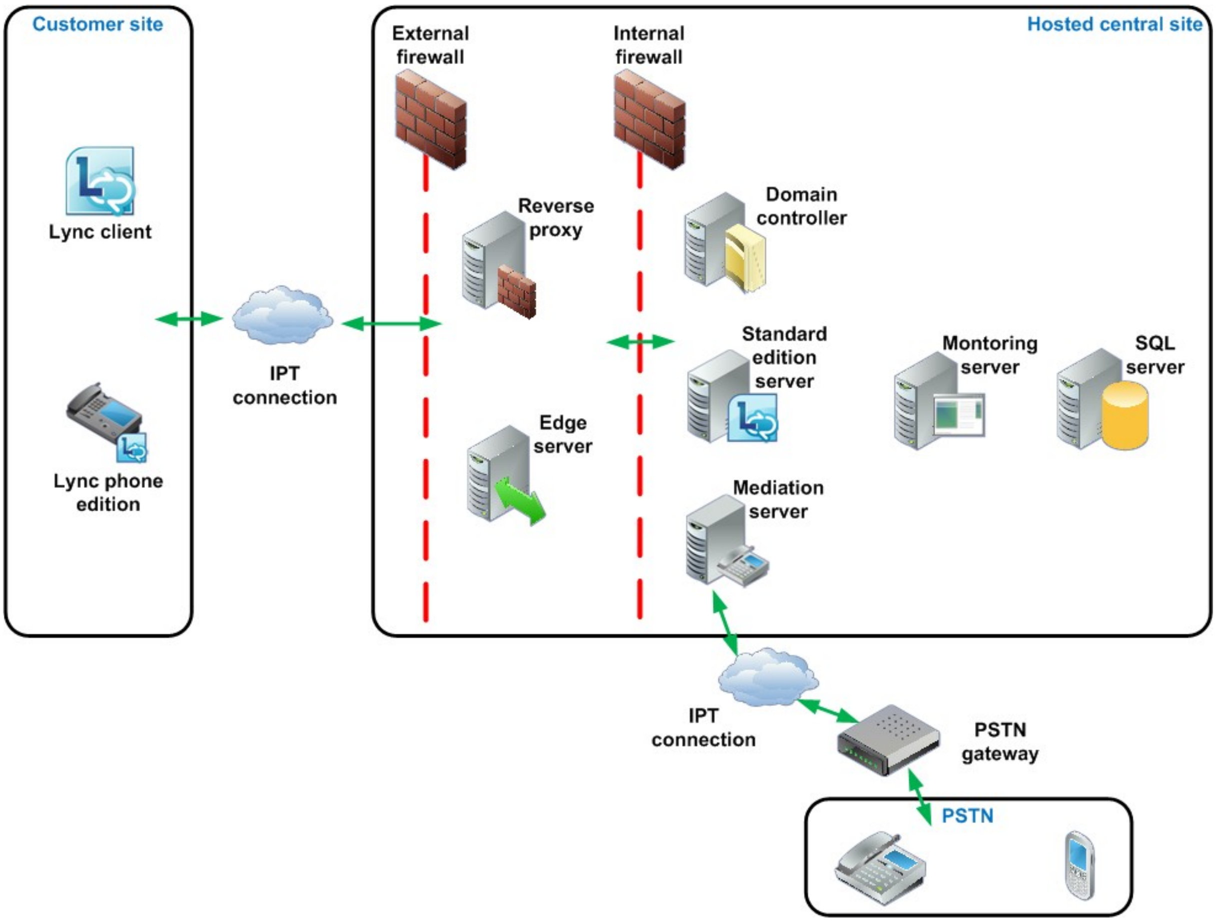

Figure 3.1: Reference architecture for hosted Lync deployment.

3.1.2 Hardware

The hosted Lync deployment is built using components from Cisco Unified Computing System [8] (UCS) architecture. The unified architecture consists of a network of connected compute nodes and storage shown in figure 3.2. The architecture consists of blade servers that are connected to blade server chassis. Each blade server chassis is connected to a fabric interconnect using a fabric extender. A fabric interconnect can be connected to a SAN to provide increased storage for the blade servers. The architecture supports up to 40 blade server chassis on the same fabric interconnect. Each blade server chassis can take up to 8 blade servers. This allows for 320 blade servers in the same infrastructure. This makes Cisco USC a highly scalable platform. For host providers it is easy to scale out the infrastructure in order to serve more customers or increasing customer demands.

The hardware specification of the hosted Lync deployment is shown in table 3.2.

The architecture of the components is shown in figure 3.2. Blade servers are placed

Table 3.2: Hardware specification for hosted Lync instance

Component	**Description**
Blade servers	CPU: 2 x Intel Xeon E5650, 6 cores, hyperthreading Memory: 96 GB Network: 10.0 Gbit/s network interface Local storage: 2 x 15k rpm SAS disk drives
Blade server chassis	Cisco UCS 5108
Fabric extender	Cisco UCS 2104xp
Fabric interconnect	Cisco UCS 6140
Storage	Central SAN connected to the fabric interconnect. The SAN is divided into disk groups consisting of 20 disks. The SAN is using RAID 60 for improved performance and redundancy.

into server chassis. Server chassis are connected to the fabric interconnect via fabric extenders.

3.1.3 Virtualization

The VMware hypervisor is used as a virtualization platform. Each blade server (also called host) is running separate instances of ESXi 5.0. VMware vCenter is used to manage all ESXi hosts. Several hosts can be organized into resource pools which serves one or more virtual machines per hosts. VMs can move from host to host within the resource pool and automatic monitoring can be enabled to maintain resource balancing among hosts. VMs are then automatically migrated from a saturated host to a new host with enough available resources. In this way virtualization can provide a lot of flexibility along with better hardware utilization. However there are several known performance degrading factors associated with virtualization.

The overhead in the VMware hypervisor is related to several individual factors [22]:

- With software-based CPU virtualization, the guest application code runs directly on the processor, while the privileged code must be translated before running on the physical processor. This is to avoid using root mode which is reserved for VMkernel. The translated code is slightly larger and executes slower. Programs with little privileged code is almost unaffected, but programs with a larger privileged code component, such as system calls, traps or page table updates can run slower in a virtual environment.
- Certain processors offer hardware-assisted CPU virtualization. This is supported by modern processor architectures, such as Intel(R) Xeon(R). Hardware-assisted CPU virtualization removes the overhead of translating the code since the processor offers a guest mode where all guest code is run. Still, on certain events (i.e. updating page tables), the processor must exit guest mode and enter root mode. These exits are handled by the hypervisor, and again, it consumes CPU resources. The number of exits and the total time spent in exits determines the

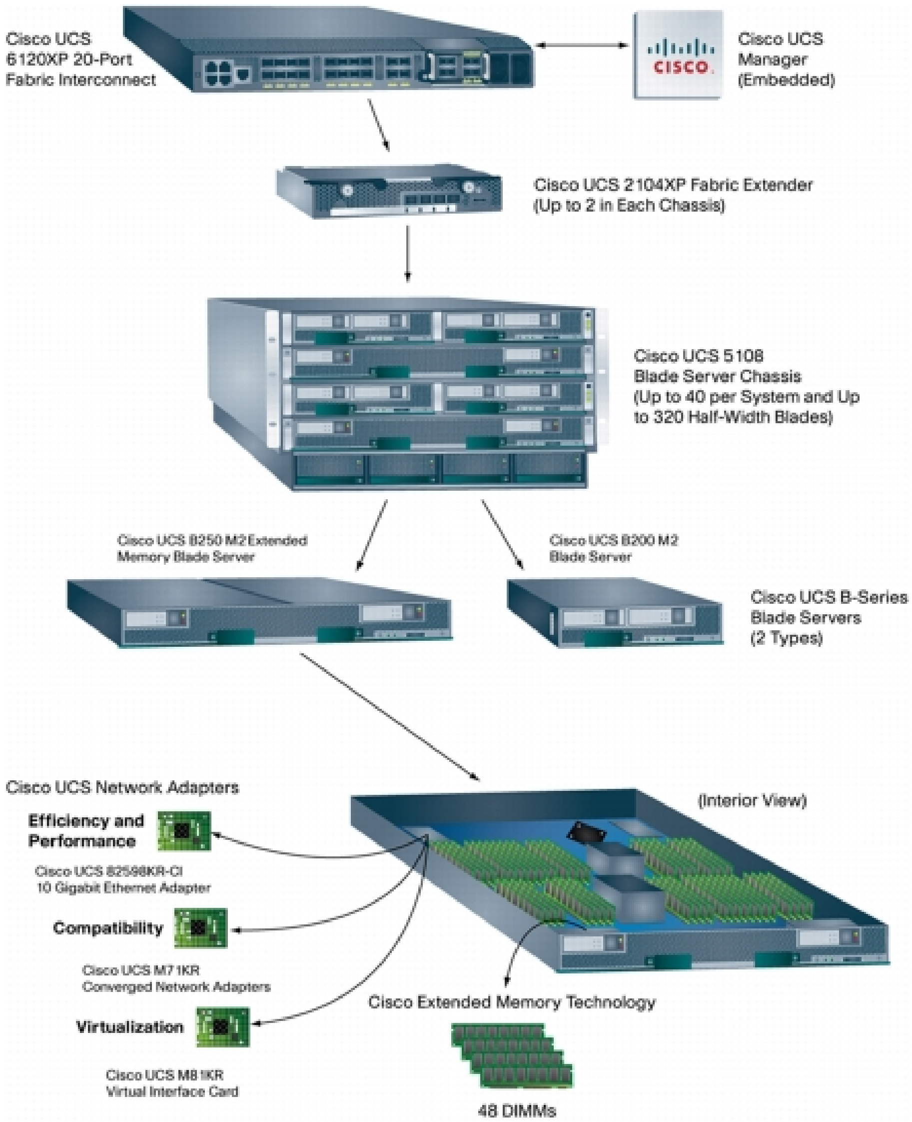

Figure 3.2: The Cisco unified computing architecture. [8]

total overhead. Software- and hardware-based CPU virtualization never coexist. The architecture studied in this report supports hardware-assisted CPU virtualization.

- The VMkernel as well as the VMconsole need prosessing time on pCPU to function properly. In a memory-bound environment with lots of CPU-wait, this overhead is most likely negligible, but in a CPU-bound environment it can affect performance.

- Processors supporting hyperthreading provides more than one, typically two, threads to run concurrently on the same processor core. Each thread can represent a vCPU. Two vCPUs sharing the same core, get only half of the shared resources (such as caches) so performance might be very dependent on the cache-hit rates.

- When VMs use physical memory a double lookup is needed. The OS has its own virtual memory which is mapped to (what the VM thinks is) physical memory.

This must again be mapped to the actual machine memory. A shadow lookup table can be used to decrease lookup wait time, but at the cost of maintaining more lookup tables. This extra lookup table consumes RAM.

- Some processor architectures support hardware-assisted memory virtualization where two layers of page tables are supported. This eliminates the cost of maintaining the shadow table in software, but the TLB miss latency is higher when using hardware-assistance.
- Both VMkernel, the VMconsole and each VM needs to allocate and use memory. This is all related to a virtualization overhead. The more vCPUs a VM has and the more memory is allocated, the more memory must be used for virtual management.
- The Virtual Machine Monitor (VMM) uses a queue to hold I/O requests before sending them to the storage device (local disk device, DAS, SAN, etc). Request from this queue may need to be reorganized to fit the start-time fair queueing (SFQ) algorithm. Some requests must also be split before departure and rejoined on arrival. Thereby I/O requests can expect a larger response time compared to bare machine performance. There is also need for CPU and RAM to maintain and process an extra level of queuing.
- Most hypervisors allow virtual network components (hubs, switches, routers, etc). These are emulated in software and requires resources to run. This adds overhead compared to bare machine performance where these devices could be represented by hardware components. The Cisco UCS [8] technology provides some virtual networking functionality that reduces the need for software emulation.

Even though virtualization introduces a lot of overhead there are also one example where the resource usage decreases compared to bare machine performance. VMware esx provides a proprietary transparent page-sharing technique that detects identical memory pages used by several VMs. This proves useful when several VMs use the same OS, shared libraries, etc. Pages containing shared libraries and static OS components are normally locked to read-only mode and several VMs can therefore safely access them and avoid keeping their local copy. This allows for memory overcommitment where memory entitlement is larger than available memory. VMware has measured up to 30 % memory usage reduction caused by this technique. The Lync servers are all installed on Microsoft Windows Server 2008 R2 and thereby shares a common operating system. This will clearly lead to reduced memory requirements in a virtualized Lync deployment.

Scaling system size

Since this report studies a Lync deployment which is installed with a standard edition licence, the *scale out* option is not possible and a *scale up* method is used instead. The maximum system size was limited by the resources available. In this test setup, only three Cisco UCS 200 M2 blade servers was available for running the virtual Lync servers and load generators. The load generators required twice the hardware resources of the Lync servers, so only on blade server was used for virtual Lync servers.

The system size was therefore scaled by reducing the resources available on the blade server. The CPU resources can be scaled by disabling processors or processor cores. As mentioned in the section 3.1.2 the blade servers uses two Intel Xeon E5650 with 6 cores and hyper-threading support. This gives a total of 24 virtual cores that are all enumerated by the hypervisor. The processor layout and virtual cores is shown figure 3.3. Each circle represents a physical processor core, and the numbers within each circle denotes the virtual processor index which is two per physical core because of hyperthreading. By using CPU affinity setting in VMware vSphere client, it is possible to decide which virtual cores can be used to schedule each vCPU of a VM. For the 2/4 system size, the affinity setting for all VMs was set to 0-11. This effectively disabled one of the two processors. For 1/4 system size the affinity was set to 0-5, so that only half the cores of one processor could be used. It is worth noting that while the number of cores are constrained, the processor caches are still widely available. The level 1 and 2 caches are separate for each core, so these scale according to the number of cores, but the level 3 cache is shared among all 6 cores of one processor. This means that for 1/4 system size, the number of cores is correctly scaled, but the level 3 cache is scaled 2/4, so it is twice as big as it should be. This gives a bias towards better performance estimates because more instructions and data can be stored in level 3 cache instead of being fetched from main memory which has a larger access time. Network resources were not scaled in this test setup. Each virtual machine has access to the 10.0 Gbit/s network interface of the blade server. Still, most of the communication between VMs are internal to the server. Only edge server, reverse proxy and mediation server need to use the network interface. Memory resources or storage resources were not scaled. Table 3.3 summarizes the hardware scaling.

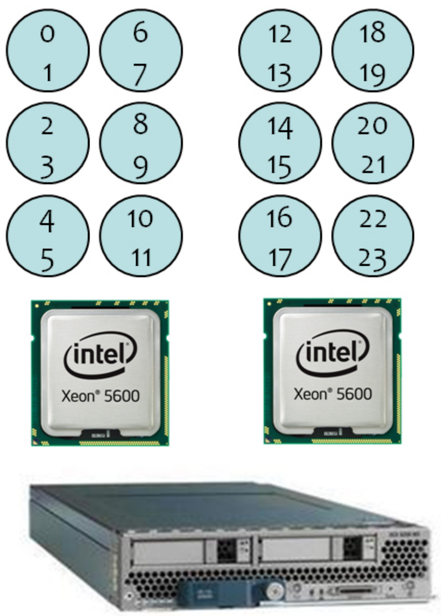

Figure 3.3: Processors and virtual cores in a blade server

Table 3.3: Scaling hardware resources

System size	CPU resources	Network resources	Memory resources	Storage resources
4/4	4/4	4/4	4/4	4/4
2/4	2/4	4/4	4/4	4/4
1/4	1/4*	4/4	4/4	4/4

* The level 3 shared cache is scaled 2/4

Tuning virtual resources

When several virtual machines are running on the same physical host they will share resources with each other. When deploying virtual machines it is important to make sure each virtual machine has the resources it needs to operate well. This section describes the hardware allocations used in the test setup. This description can help others create similar tests to verify results. An overview of resource allocations for each virtual machine is shown in table 3.4.

Table 3.4: Virtual resource allocation

Virtual machine	Disk drive	Memory	vCPUs 1/4 size	vCPUs 2/4 size	vCPUs 4/4 size
Front end	40 GB	32 GB	4	7	14
Edge	40 GB	16 GB	2	4	7
SQL	100 GB	32 GB	1	2	3
Mediation	40 GB	16 GB	1	1	2
Monitoring	40 GB	16 GB	1	1	1
Reverse proxy	40 GB	16 GB	1	1	1
Domain contr.	40 GB	16 GB	1	1	1

The disk drive size of most virtual machine is 40 GB, which is the default allocation size for Microsoft Server 2008 R2 installations. At first, the SQL server was also provisioned with this storage amount. During the first tests with heavy load, the disk ran full and performance was seriously degraded. The SQL server stores call detail records which are generated for each ongoing audio session. With heavy load a lot of data needs to be stored to the database. Therefore the disk size was increased to 100 GB.

Memory is provisioned to each virtual machine based on the hardware requirements for similar physical machine configuration recommendations from Microsoft [14]. The front end server and SQL server which both contains database entities are provisioned with 32 GB of memory while the other machines get 16 GB. More memory is beneficial for database servers, because keeping a large portion of the database in memory will reduce the overhead of reading from disk. As shown in the results chapter all VMs are provisioned with enough memory. The load on these virtual machine Lync servers are lower than the physical machine equivalents which are described in Microsoft's hardware recommendations.

There are several allocation options for provisioning CPU resources.

1. **Alloaction shares:** Shares specify the relative importance of a virtual machine (or resource pool). If a virtual machine has twice as many shares of a resource as

another virtual machine, it is entitled to consume twice as much of that resource when these two virtual machines are competing for resources. Default share values are (1) high - 2000 shares per vCPU, (2) normal - 1000 shares per vCPU and (3) low - 500 shares per vCPU. A custom share value may also be configured for each virtual machine.

2. **Allocation reservation**: A reservation specifies the guaranteed minimum allocation for a virtual machine. The reservation is expressed in concrete units (megahertz or megabytes).

3. **Allocation limit:** Limit specifies an upper bound for CPU resources that can be allocated to a virtual machine. Like reservation it is expressed in concrete units.

The VMs in the test setup were provisioned using allocation shares. Each VM had a normal allocation setting (1000 shares per vCPU), so the actual CPU allocation was defined by the number of vCPUs allocated to each VM. The number of vCPUs that were allocated for each VM was calculated based on the relative utilization of each VM and the number of available virtual processors. This is shown in the following formula:

$$vCPU_{VM} = \frac{\%RUN_{VM}}{\%RUN_{total}} \times N_{logproc} \tag{3.1}$$

As shown in the previous section, the number of available logical processors is 6 for 1/4 system size, 12 for 2/4 and 24 for 4/4. Table 3.5 shows the calculation of vCPUs for each VM and each system size. Theoretically each VM could have allocated the maximum number of vCPU. However, overprovisioning CPU resources may degrade performance. The VMware CPU scheduler uses an algorithm called relaxed co-scheduling. The term co-scheduling refers to the fact that all vCPUs belonging to the same VM should be treated as a group (this is of course as long as more than one vCPU is allocated). Therefore, the hypervisor tries to schedule all vCPUs in a group at the same time. Non-continious scheduling of vCPUs would create a different processor experience for the VMs than they would otherwise experience on a physical processor. SMP programs that uses several threads to complete a compute intensive task, may rely on synchronization among threads. If some threads are halted because a vCPU is not scheduled, all other threads (and their respective vCPUS) will suffer a delay. The VMware hypervisor therefore implements a co-scheduling algorithm that tries to assign equal time intervals to each vCPU. If a certain vCPU has received more processing time than the other vCPUs in the same group, it is put into a waiting state called CO-WAIT. *Relaxed* co-scheduling means that a vCPU must succeed a certain time threshold in order to be put in the CO-WAIT state. More vCPUs per group increases the chances of one or more vCPUs to be put into CO-WAIT state. Spending time in CO-WAIT state decreases performance. The vCPU calculation in (3.1) is an attempt to reduce the CO-WAIT overhead.

Table 3.5: Virtual CPU allocation

VM	1/4 all.	1/4 util.	2/4 calc.	2/4 all.	2/4 util.	4/4 calc.	4/4 all.
Front end	4 vCPU	232.301	6.79	7 vCPU	412.822	13.19	14 vCPU
Edge	2 vCPU	103.698	3.03	4 vCPU	200.972	6.42	7 vCPU
SQL	1 vCPU	38.353	1.12	2 vCPU	76.253	2.44	3 vCPU
Mediation	1 vCPU	18.797	0.55	1 vCPU	35.383	1.13	2 vCPU
Monitoring	1 vCPU	11.827	0.35	1 vCPU	18.608	0.59	1 vCPU
Reverse proxy	1 vCPU	4.777	0.14	1 vCPU	5.706	0.18	1 vCPU
Domain contr.	1 vCPU	1.034	0.03	1 vCPU	1.298	0.04	1 vCPU

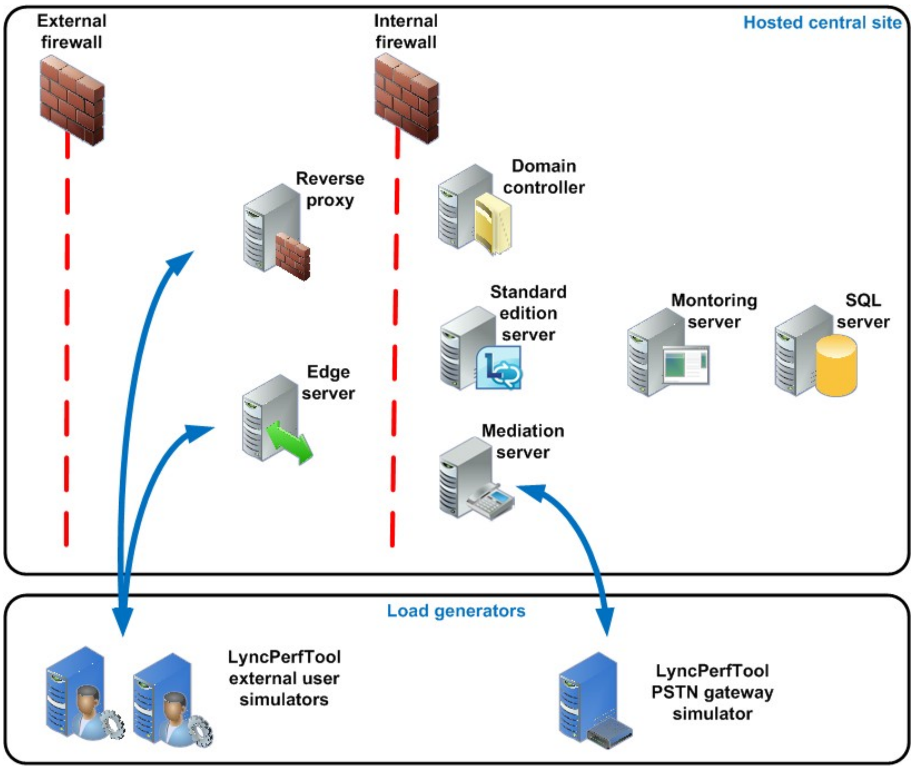

Figure 3.4: Generating load using LyncPerfTool.

3.1.4 Generating load

Lync Server 2010 Stress and Performance Tools [3] (LyncPerfTool) was used to create artificial load used for the Lync servers. LyncPerfTool is designed by Microsoft to generate realistic load for performance evaluation or stress testing of any Lync environment. LyncPerfTool should be installed on dedicated hardware so that it doesn't interfere with performance measures on Lync servers. The tests performed in this report uses two Cisco UCS M200 blade servers for running the LyncPerfTool clients. The configuration is shown in figure 3.4. In order to obtain good provisioning guidelines the test setup should be similar to an actual deployment. There are three drawbacks with the test setup that may create bias in the performance measurements.

1. **Uncorrect network distance:** The LyncPerfTool external user simulators are connected to the same LAN as the reverse proxy and edge server. The PSTN gateway simulator is also connected to the same LAN as the mediation server. In a real

deployment both clients and the PSTN gateway would be connected by an IPT connection. The IPT connection would introduce some network latency and possibly network congestion that can not be simulated in the test setup. The reduced latency and congestion will create a bias towards better performance estimates for the test setup. The bias can be reduced if IPT connections are provisioned to avoid network congestion and thereby also reducing latency.

2. **Lack of wireless networks:** In a real deployment users at a customer site may use wireless networks. Wireless networks are popular in office environments and home offices because no network cabling is needed. However, a wireless network will introduce significantly more network delay compared to a switched cable network. There may also be a larger packet loss rate on wireless networks [24]. The latency effect of wireless network cannot be simulated in this test and this will also create a bias towards better performance estimates. The bias may be reduced if wireless networks are prohibited or if wireless network implement QoS prioritizing for real time protocols.

3. **Unencrypted protocols:** In Telenor's hosted Lync setup all traffic is encrypted using either TLS or SRTP. In the test environment with simulated load, almost all traffic is encrypted using the same protocols, except the traffic to and from the PSTN gateway simulator. This is due to a limitation in LyncPerfTool that doesn't support encrypted protocols. Details on this issue is found in appendix C. The substitute protocols are TCP for SIP signalling and RTP for audio streams. Encryption adds an extra layer of complexity and processing requirements to the network protocol. The lack of encrypted protocols will create a lighter load that will create a bias towards better performance estimates.

LyncPerfTool can simulate all the main user load types, but has some constraints. An overview of load types are presented in table 3.6. There are three workload types that are not supported by LyncPerfTool. The following three sections gives a brief discussion on each of these workload types in order to evaluate the impact on the test results.

Table 3.6: User load types supported in LyncPerfTool [3]

Supported	**Not supported**
Instant messaging (IM) and presence	Group Chat Console
Audio conferencing	Web conferencing
Application sharing	Video load for peer-to-peer calls or conferencing
Voice over IP (VoIP), including public switched telephone network (PSTN) simulation	
Web Access Client conferencing	
Conferencing Attendant	
Response Groups	
Distribution list expansion	
Address book download and address book query	
E911 calls and Location Profile	

The group chat functionality enable multiple users to participate in conversations in which they post and access content about specific topics. This is somewhat similar to an IM conference, but the content of each session can be persistent, which means it continues to be available after a session ends. This means that people from different locations and departments can participate, even when they are not all online at the same time. Enabling group chat requires installation of special group chat communications software and back end database on separate servers. The group chat functionality is not a part of Telenor's standard hosted configuration. Therefore, the lacking support of this load type in LyncPerfTool will not affect the measurements.

Web conferencing includes desktop sharing, application sharing, powerpoint presentations and virtual whiteboard. While LyncPerfTool does not support this feature, it does support conferencing using the web access client. The difference between the two is that the web access client doesn't allow voice or video in the conferences. The lack of voice is not crucial because it is simulated in dedicated audio conferences. The lack of video load follows the same conclusion as the general peer-to-peer/conferencing discussion of video in the next paragraph.

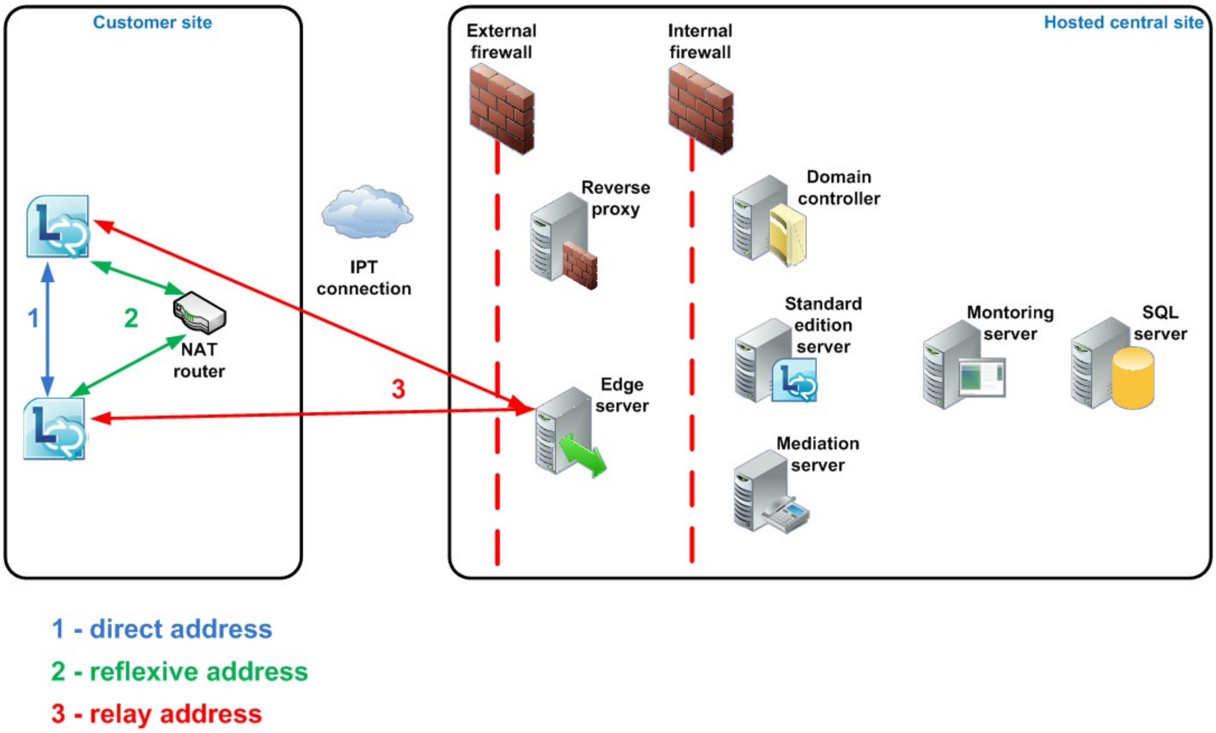

Figure 3.5: External user video stream exchange options

Video conferencing allows multiple users to join a conference and share audio/video with each others. This requires the Lync client user station to be equipped with a camera and microphone. The audio is collected from each user and mixed at the A/V server before being sent to each user. The mixing includes echo reduction, dominant speech amplification and other voice quality improvements. The video stream however does not require any central processing and is sent directly from one user to another. Each user can switch camera view, but can only view one camera at the same time. This means that audio streams need processing at the A/V server while video streams are only forwarded. Lync uses the Interactive Connectivity Establishment (ICE) [16] protocol to negotiate on a shortest possible network path before initiating a video stream. For external users there are three different alternatives to exchange video streams peer-to-peer, see figure 3.5. The first option is to use the direct address between the two Lync clients. This will always be the shortest and fastest option. However if client firewalls prohibit such communication or the clients are not on the same network (e.g. one client can be at a home office and the other on a corporate network) this option is not possible. The second option is to connect using the reflective address of the NAT router. If this option is somehow prohibited the clients can also negotiate on a third alternative, relaying the video session through the edge server external network interface. If all users are internal or all users are external, video streams can be sent peer-to-peer with the direct or reflective address option. However if users are both internal and external, they need to relay video streams through the edge server. This alternative requires the edge server to actively forward each packet in the video session. In the hosted Lync scenario there are 100 % external users. This means that the lack of video simulation in LyncPerfTool will not create any bias in performance measures as long as we assume that users in a real deployment can use either the direct or reflective address connection option.

3.2 Scalability modelling

When considering provisioning of hardware resources it is important to determine the scalability of the system. As discussed in the introduction chapter, scalability describes the relationship between system capacity and system size. Optimally this relationship is linear, but often it is sub-linear. A scalability model can help understand the causes of sub-linear scaling as well as determining when they occur. This report uses the Structure and Performance specification method SP [17] to model Lync servers. The SP framework models the interactions between operations on software components and hardware resource demands. Resources are defined in three dimensions; processing, network and storage. Each software component can have several operations. An operation on one component may require operations from another component. Complexity matrices are used to specify the interactions among each component. A generic SP model for a web server with an SQL back-end is shown in figure 3.6. This figure can help explain the basics of SP modelling. The bottom components represent hardware resources like network, CPUs and disk drives. Memory, processing and communication links are denoted by bold, solid and dashed lines respectively. The model shows that the user browser uses the LAN to communicate with the web application which in turn communicates with the MySQL database. The web application can be decomposed into a web server component and a web server OS component which all get processing resources from the same web CPU. Readers are referred to [17] for a more detailed description of the SP model.

Constructing an SP model of a system consists of four main steps:

1. Identify software components and hardware resources. An acceptable level of details must also be chosen.

2. Operations on each component must be identified and described.

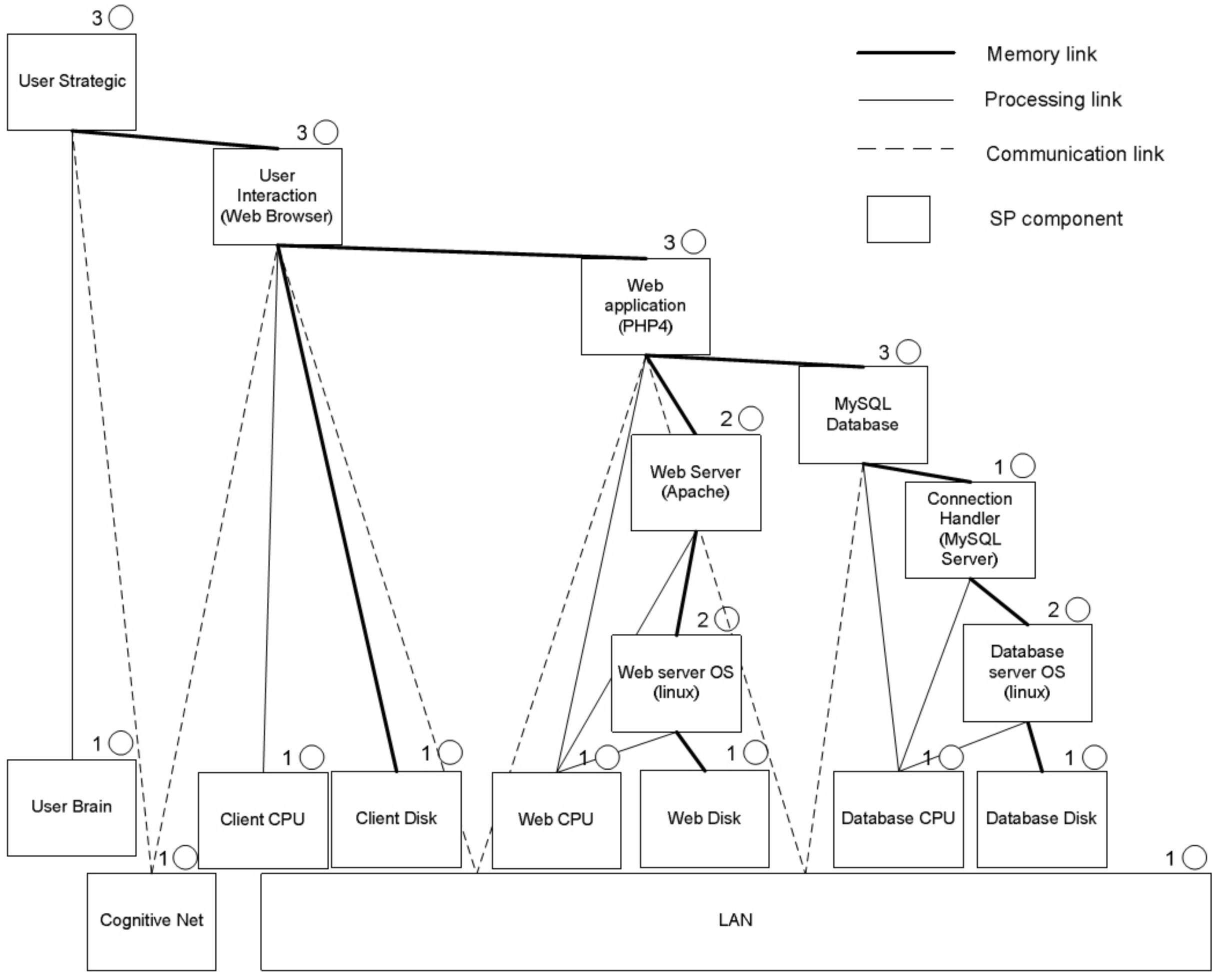

Figure 3.6: A generic SP-diagram for a web server system

3. The relations between operations on different components must be described and quantified by populating complexity matrices. These matrices can consist of integer values as well as functions.
4. Measure the resource demand of each operation on every hardware component.

In order to complete the first two steps in the above list a detailed description of the system is needed. Three different resources were used to obtain this information: (1) Microsoft's online documentation of Lync [13], (2) network captures from the test environment and (3) Lync server logs from load simulations. The online documentation focus on describing the different server roles, how they interact and how they should be configured. It is written to help system administrators configure the system rather than helping a performance analyst understand every detail of the system.

When comleting step one of building the SP model, it was natural to use the servers and LyncPerfTool clients as main software components. Theoretically each server could be further analyzed and divided into several sub components, but obtaining this kind of information would require more knowledge on the processes within Lync. The network captures and server logs can easily be analyzed on a per-server basis, but provide little information on distinct processes within each server.

The domain controller is not included in the model. This server role does not provide any services that directly relate to the user load from LyncPerfTool clients.

Each server has access to processing, network and storage resources. The mediation server communicate over the LAN to the LyncPerfTool PSTN GW. The reverse proxy and edge server communicate to the LyncPerfTool clients. Other than these three servers, all communication among servers are internal on the ESXi host, i.e. it uses an internal bus for communicating among servers. Therefore, the SP-diagram shows that a lot of servers don't use the LAN. The internal bus could be modelled, but the resource demand of an internal bus is by far negligible when compared to a LAN. This simplification on network usage reduced the complexity of the model. All servers need processing resources to carry out a variety of operations. When it comes to hard drive usage, most Lync servers uses it for reading instructions, logging, etc. The front end server and SQL server contain database instances that require a lot more reading and writing. However, when these servers are provisioned with enough memory to keep the whole database in-memory, the dependency on hard drive resources will decrease.

Figure 3.7 shows the SP-diagram of the hosted Lync deployment. The diagram shows how step two of the modelling process is implemented. All operations are initiated by either LyncPerfTool client or PSTN GW simulator. Remote users must connect to the edge server which authenticates users and forwards traffic to the front end server. Phone users communicate over the PSTN to the mediation server which translates audio sessions and receives SIP signalling. While remote users can initiate a variety of operations which cause load to the central site, phone user can only use audio calls or audio conferencing.

The lower level hardware resources has simple operations. The hard drive resources provides read and write operations, while the processor has a single execute instruction operation.

The third and forth step in creating the SP model requires finding the relations among the different operations on each server and measuring the resource demand of each lower level hardware operation. Complexity matrices are used to represent the relations among servers. The entities within the matrices are mostly integers. E.g. one *IM message* operation on LyncPerfTool client process would require one *forward IM message* operation on the edge server, which in turn would require one *reflect IM message* operation on the front end server. Other operations such as audio conferencing may not give integer entities in the matrices. Audio conferencing requires processing of several audio streams at the front end server, one for each participant in the audio conference. The server has to do some processing for each ongoing audio conference and some extra processing for each extra participant on the conference. This gives more complex matrix instances which are harder to determine.

When it comes to measuring the resource demands this is normally done by the following protocol: (1) Measure the average resource usage on an "empty" system, i.e. a system with no user operations. (2) Measure the average resource usage when performing operation X on a given server. (3) Calculate the difference between operation X and the "empty" system measure and divide by the number of X operations carried out. This gives the resource demand for a single operation and should be repeated for all operations. Using this measurement protocol requires the ability to simulate operations individually and measuring the corresponding resource demands. However,

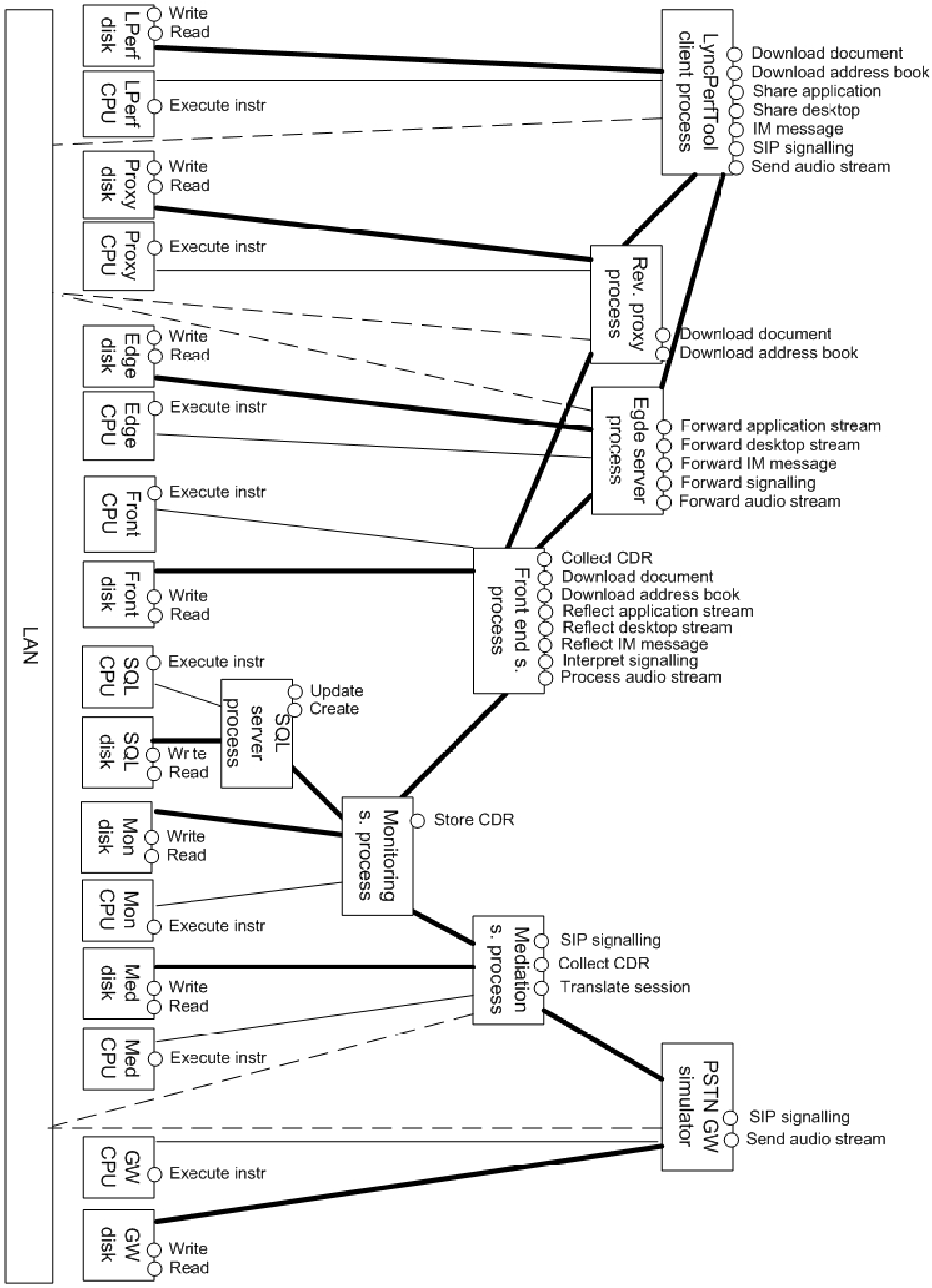

Figure 3.7: SP-diagram for Microsoft Lync 2010

when all load is generated using LyncPerfTool, it is impossible to run each operation in isolation. E.g. an audio conference would create both audio streams as well as SIP messages that are used for signalling. The front end server will process ongoing audio sessions while also creating CDR records to send to the monitoring server. It would not be possible to create realistic SIP messages or CDR records without also simulating a full audio conference. Measuring the resource demand would give a combination of several operations that would be hard to separate.

Experience shows that performing the two first steps of the SP modelling was feasible, while the third and forth steps were almost impossible given the available tools for load generating and performance measuring. Therefore it was not possible to complete the SP model and consequently not possible to use the SP model to predict scalability of the Lync deployment. Without a proper scalability model, the scalability analysis had to be carried out using empirical measures only.

3.3 Empirical scalability tests

While the static scalability modelling approach turned out to be rather hard, scalability can also be determined using empirical scalability tests. The drawback of using empirical scalability tests only is that the scalability is only determined for the same system size range that is actually measured.

3.3.1 Binary search method

The objective of the empirical tests is to determine the maximum number of users for each system size and user load scenario. A number of quality metrics are used to assure that the quality degradation of the Lync services are within acceptable thresholds. The maximum number of users can be found by trial and error and adjusting the number of users for each run. The number of users are increased if quality metrics are within thresholds, and decreased if the thresholds are exceeded. The most effective way of reducing the remaining search space is to use a binary search. For each iteration of the binary search the search space is halved. Interpolation on quality metrics could also be considered, but since there are no guarantees on linearity in the metrics, a binary search is a more robust alternative. The binary search can theoretically continue until the search space has only one remaining user. However, searching with such fine granularity would probably not produce stable results because of the random errors in all measures. Such errors are more dominant in fine granularity tests. Running many iterations also requires more time for testing. In these tests 4-6 iterations was used for all system sizes until a near-correct value was found.

3.3.2 Quality metrics

Quality metrics were collected from two distinct sources, the monitoring server and the LyncPerfTool clients. The thresholds in table 3.7 is found in Microsofts recommendations [5] and the Lync sizing study [19].

The front end server and mediation server automatically samples all ongoing sessions and sends CDRs to the monitoring server. The monitoring server produces statistics on audio quality using several metrics. Round trip time describes the time it takes from a request is sent to the answer is recieved. A high round trip time can be noticed by the user as a speech delay. Buffering enables acceptable speech quality even

Table 3.7: Quality metric thresholds [5, 19]

Monitoring server statistics	
Metric	**Threshold**
Round trip time (RTT)	100 ms
Jitter	20 ms
Packet loss	0.1
Mean opinion score (MOS)	0.5
LyncPerfTool performance counters	
Metric	**Threshold**
SIP 503 messages/sec	≈ 0
SIP 504 messages/sec	≈ 0

with a high round trip time. Jitter measures the variance in round trip time. A high jitter value makes buffering and processing hard because the packets arrive at highly variable times. Packet loss denotes the number of packets that are lost in transmission. The real time network protocols used in Lync has several built-in mechanisms for reconstructing the speech with acceptable quality despite some lost packets. However, when the packet loss exceed 0.1, the packet loss affects call quality. Mean opinion score (MOS) is a compound metric used to measure end user quality of experience. MOS originates from the ITU-T standardized absolute categorization scale (ACR) which is a subjective rating of call quality on a scale from one (bad) to five (excellent) [20]. The scale is built to allow persons to rate the call quality based on actual user experience. ACR is a subjective metric and MOS is created as its objective counterpart and uses the same scaling range. MOS is calculated using both transport layer parameters such as packet loss, jitter and delay as well as payload parameters such as audio codec, noise-level, echo, gain and talk-over effects [12]. Each codec that is used in Lync has a maximum MOS value [1]. The values are shown in table 3.8.

Table 3.8: Codecs used in scenarios with maximum MOS [1]

Scenario	Codec	Max MOS
UC-UC call	RTAudio wideband	4.10
Conference call	Siren	3.72
PSTN call	RTAudio narrowband*	2.95
PSTN call	Siren*	3.72

* The codec used in PC-PSTN can either be Siren or RTAudio NB depending on the configuration of the Mediation Server.

The monitoring server records the degradation of MOS value, i.e. the difference between the maximum MOS and measured MOS. The degradation should not exceed the 0.5 threshold.

From the LyncPerfTool, SIP 503 and SIP 504 error messages were counted. SIP 503 is a reply message sent by the Lync server, indicating that it is too busy to handle the request. A SIP 504 message is a server timeout message. LyncPerfTool counts the average number of such messages per second. Error messages may occur by random if the server has a short period of high utilization while the overall utilization is low. This means that some messages can be accepted, but over a longer period of time, the number should be close to zero. No discrete threshold is set, because the acceptable number of messages is influenced by the total number of users. The threshold is therefore ≈ 0 for the SIP error messages.

A number of quality metrics could also be collected from the front end, edge or mediation server. This instrumentation would however affect the performance measures and are not included in the tests. The monitoring server reports are collected after each test is run and will not affect performance. The metrics collected on the LyncPerfTool clients will not affect performance either because the clients run on dedicated hardware. This means that the instrumentation required to measure performance did not create any bias on the actual measurements.

3.3.3 Measuring resource usage

This section describes how the performance measures were collected. This includes the required tools, the necessary configuration and test method.

Esxtop performance tool

Esxtop is a command line program that is installed by default on every ESXi host. To enable this tool the ESXi console and SSH server must be configured at the ESXi host. The esxtop tool can measure the performance of VMs in isolation from the hypervisor overhead. This makes the tool ideal for these performance measures. Esxtop can be used in both interactive and batch mode. The interactive mode lets the user view a snapshot of the parameters in real-time. The batch mode can be used when several samples are collected over time and written to a log file. The sample intervals are customizable, but 2 seconds is the shortest possible interval.

Selecting power management policy

Most modern processors support Advanced Configuration and Power States (ACPI) that can reduce power consumption when the machine does not require full processor capacity. Intel Xeon has 9 discrete power states that can be selected in order to lower processor speed and power usage. VMware ESXi 5.0 has four different host power management policies to choose from [21]; (1) high performance, (2) balanced, (3) low power and (4) custom. When the high performance policy is selected, the processor always runs in the highest P-state. The other policies uses different algorithms to select between the 9 available power states. Compared to the highest processor speed, a processor running on slower speed will always require a higher utilization to perform the same operation. Running in a lower power state with a slower processor speed will therefore affect the utilization measures of esxtop. This makes it suitable for low load steady-state experiments. While the high performance policy was default in ESX/ESXi 4.0 and 4.1, ESXi 5.0 uses the balanced policy by default. Each ESXi hosts must therefore be configured to use high performance policy as long as the experiments are carried out.

Measurement method

For each iteration of the binary search a number of parameters was measured. The parameters have natural variance, so a single discrete sample of each parameter has little value. By collecting a number of samples over a longer time period and averaging the results this variance can be reduced. LyncPerfTool is designed to create a stable load. However it takes some time before the load is stable. When the load has stabilized, the system has reached a steady-state. This steady-state performance can be measured and averaged. The time period from starting the stress tool to reaching the steady state is called the transient period. Figure 3.8 gives a graphical overview of the transient interval and the steady-state measurement period. The graph shows the utilization of the mediation server in the first 95 minutes after LyncPerfTool started.

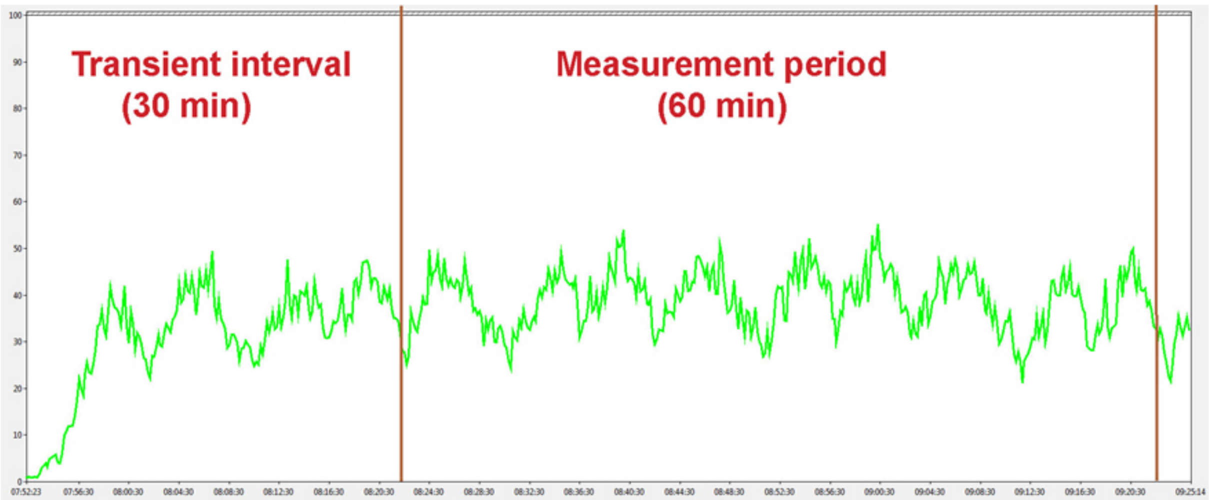

Figure 3.8: Utilization on mediation server during testing

When LyncPerfTool is initialized, it first signs in all users. This is done at a rate of 1 user per second. The different load types are run as separate processes so several users are signed in in parallel. Thereby the maximum time to sign in all users is determined by the IM conferencing load process which has the most users. For the highest number of users, 4800 users (the low load test on 4/4 system size), the number of IM users was 1587. This means that all users are signed in after 26 min and 27 seconds. When the number of users is lower, the sign in delay is also smaller. A transient interval of 30 min was therefore selected and a total steady-state measurement period of 60 min. Some of the tests could have used a shorter transient time, but standardized transient times made log parsing easier. Esxtop was started right after LyncPerfTool and ran for 90 minutes. When finding the average values, the first 30 seconds where excluded. The following esxtop configuration was used for all tests.

```
# esxtop -b -d 2 -n 2700 > output.csv
```

```
-b:     batch mode
-d N:   sampling period
-n N:   number of samples
```

With a sampling period of 2 seconds, 2700 samples are needed to make the measurement last for 90 minutes ($\frac{90 \times 60 seconds}{2 seconds/sample} = 2700 samples$):

Chapter 4

Results

Empirical testing was performed by the protocol described in the method chapter. Binary searches were carried out for the $1/4$, $2/4$ and $4/4$ system sizes. The complete results, which cover several pages, are presented in appendix B and this chapter only outlines the main findings.

4.1 Performance parameters and quality metrics

For each of the different system sizes and user load scenarios, the maximum number of users was found. Table 4.1 shows an overview of all performance parameters that was measured for each scenario. The CPU group %RUN values are normalized for easier comparison. The table shows that all quality metrics are within the given thresholds.

4.2 Bottleneck resource

This section gives an evaluation of all four resource types considered and concludes on the bottleneck resource. The bottleneck is the resource type with the highest utilization.

Memory

All servers where provisioned with memory resources according to Microsoft's server recommendations. If any of the servers had too little memory, they would need to start swapping memory to disk. Table B.1 - B.6 shows that the memory swapping was always 0. This means that memory was not a bottleneck.

Table 4.1: Performance metrics for all binary search results

System size	1/4	1/4	2/4	2/4	4/4	4/4
User profile configuration	High	Low	High	Low	High	Low
Number of users	350	1000	650	2000	1250	3900
Lync server esxtop statistics: Group CPU						
% RUN front end	58,075	58,903	58,974	59,126	59,427	59,653
% RUN edge	51,848	52,432	50,243	50,643	52,434	52,613
% RUN SQL	38,353	34,29	38,1265	38,265	43,173	43,281
% RUN mediation	18,797	15,925	35,383	35,915	34,342	34,592
% RUN monitoring	11,827	11,091	18,608	18,068	26,897	27,017
% RUN rev. proxy	4,777	4,589	5,706	5,756	6,65	6,691
% RUN domain contr.	1,034	1,091	1,386	1,294	1,427	1,364
Lync server esxtop statistics: Memory						
Swap Mbytes Read/sec	0	0	0	0	0	0
Swap Mbytes Read/sec	0	0	0	0	0	0
Lync server esxtop statistics: Network						
Mbit Rec/sec mediation	0,878	1,787	1,834	3,587	3,245	6,989
Mbit Tra/sec mediation	0,832	1,760	1,53	3,300	2,612	6,249
Mbit Rec/sec edge	13,212	26,668	23,689	50,670	44,621	95,514
Mbit Tra/sec edge	220,351	40,341	35,129	76,238	73,561	158,564
Mbit Rec/sec rev.proxy	0,007	0,016	0,015	0,034	0,032	0,061
Mbit Tra/sec rev.proxy	0,014	0,024	0,022	0,046	0,043	0,094
Lync server esxtop statistics: Physical disk						
Average driver ms/comm	5,068	5,071	5,089	5,091	5,724	5,716
Average kernel ms/comm	0,015	0,015	0,015	0,015	0,15	0,015
Average guest ms/comm	5,093	5,096	5,114	5,116	5,74	5,905
Lync PerfTool.exe esxtop statistics: Group CPU						
% RUN client 1	0,353	0,401	0,524	0,598	0,751	0,854
% RUN client 2	0,354	0,402	0,525	0,604	0,753	0,855
% RUN PSTN gateway	12,684	17,1852	19,196	26,841	27,369	36,219
Lync PerfTool.exe performance counters						
Local 503 responses/sec	0	0	0	0	0	0
Local 504 responses/sec	0	0	0	0	0	0
Monitoring server statistics						
Round trip time UC conf	39	23	42	36	42	25
Jitter UC conf	13	5	6	7	7	5
Packet loss UC conf	0,05	0,01	0,03	0,04	0,04	0,03
MOS degradation UC conf	0,41	0,43	0,45	0,47	0,44	0,47
Round trip time UC leg	32	20	13	10	9	8
Jitter UC leg	13	4	7	4	4	3
Packet loss UC leg	0,06	0	0,04	0,03	0,04	0,03
MOS degradation UC leg	0,5	0,41	0,23	0,26	0,41	0,38
Round trip time GW leg	5	3	3	2	2	3
Jitter GW leg	4	1	1	1	2	1
Packet loss GW leg	0	0	0	0	0	0
MOS degradation GW leg	0,19	0,19	0,12	0,15	0,09	0,13

Network

All servers had access to a 10.0 Gbit/s network interface. However, only the mediation server, edge server and reverse proxy server needed to use this interface. Table B.1 - B.6 shows that the highest network utilization of all servers combined where 267 Mbit/s. Not including hypervisor overhead this gives a total utilization of 2.67%, far below the utilization of the processors. Network resources was therefore not a bottleneck in this Lync deployment.

Disk

The disk utilization was evaluated by observing the time required to complete disk operations. As long as this time is kept below 15 ms, there is no contention on disk resources [19]. The highest observed disk operation time was 5.903 ms. This indicates that there was not a high enough utilization of disk resources to create contention. Disk resources was not a bottleneck.

CPU

The utilization of CPU resources has a close correlation to system performance. Table 4.1 shows that the utilization of the front end server ranges from $58 - 60\%$ for all maximum users measures. The processor is the highest utilized and should therefore be considered the bottleneck in this Lync deployment.

4.3 Evaluating quality metrics

All quality metrics has different thresholds and units. A normalization is required to compare the different metrics. In figure 4.1, all parameters from the monitoring server are normalized by dividing the measured value by the threshold. This means that when the curve crosses the horizontal line marked 1, the threshold is reached. The results come from the 4/4 system size, high user load scenario, but the shape of the metrics curve is representative for all scenarios and system sizes.

The figure shows that the MOS metric is the first metric that crosses the threshold. System administrators that monitors the system to detect and mitigate any performance issues wants to have an early warning on system contention. Thereby, MOS value as the best candidate for monitoring a single metric. The other metrics could also be monitored, but they would give a more postponed warning.

Figure 4.1 does not include the performance counters from the LyncPerfTool clients. These counters were always 0 when the highest possible number of users were reached. Small deviations of up to 0,04 SIP error responses/sec were observed when the system was highly over-utilized. The SIP error messages are not considered a good metric for monitoring system contention. The monitoring server metrics consider the real-time performance of ongoing voice sessions, etc. Real-time traffic is especially vulnerable to

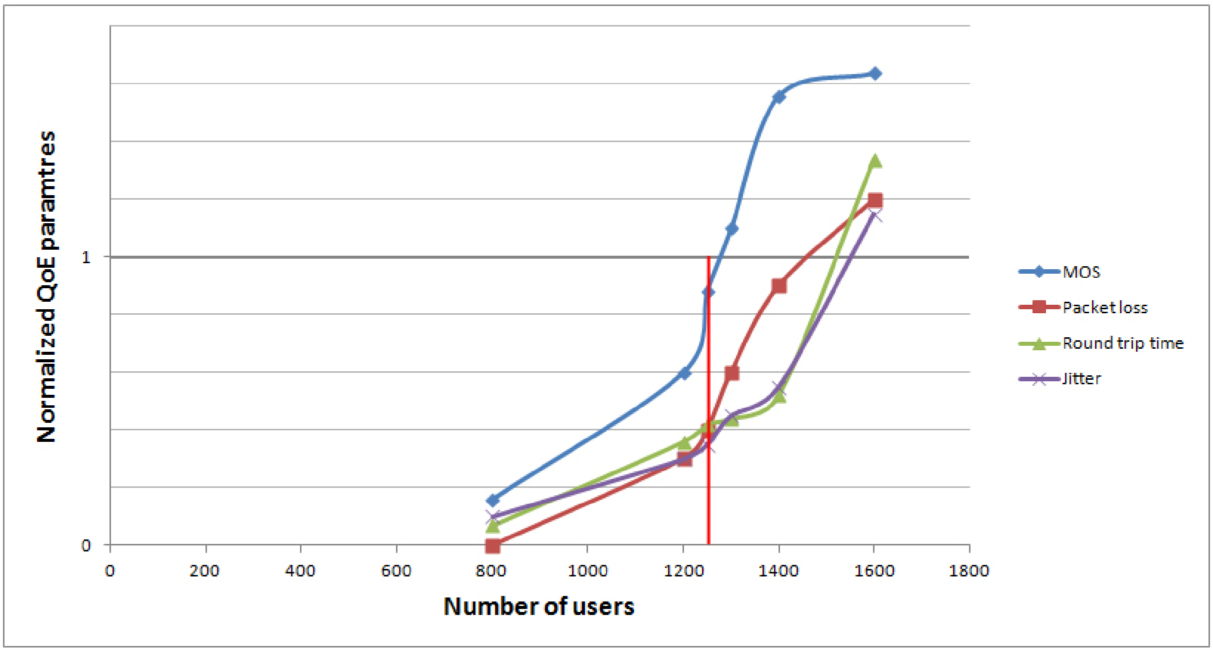

Figure 4.1: Normalized QoE parameters, from the 4/4 system size high user load test

system contention and this may be the reason why the quality metrics gave an earlier warning compared to SIP error messages.

4.4 Evaluating utilization

Section 4.2 shows that CPU is the bottleneck resource. A closer look at the utilization of each VM may give more insight into how the system behaves under different CPU loads. Figure 4.2 shows the normalized CPU utilization values for all servers in the 4/4 system size, high load user scenario. The red vertical line shows the maximum number of users (1250). From the figure it is possible to observe a breakpoint in the front end server utilization curve when maximum number of users is met. Similar break points are also observable in other servers, such as the edge server. But on the edge server, the break point comes after the maximum number of users is reached.

From table 4.1 it is clear that the utilization of the front end server is always below 60% for all measurements where the quality metrics are within thresholds. For all other measurements, the utilization is above 60%. It is therefore reasonable to say that monitoring the utilization of the front end server would be a good indicator on system health. This measure could be used by system administrators. They could even set an alert at 50% utilization just to get an early warning on possible performance issues and thereby having the time to mitigate the consequences.

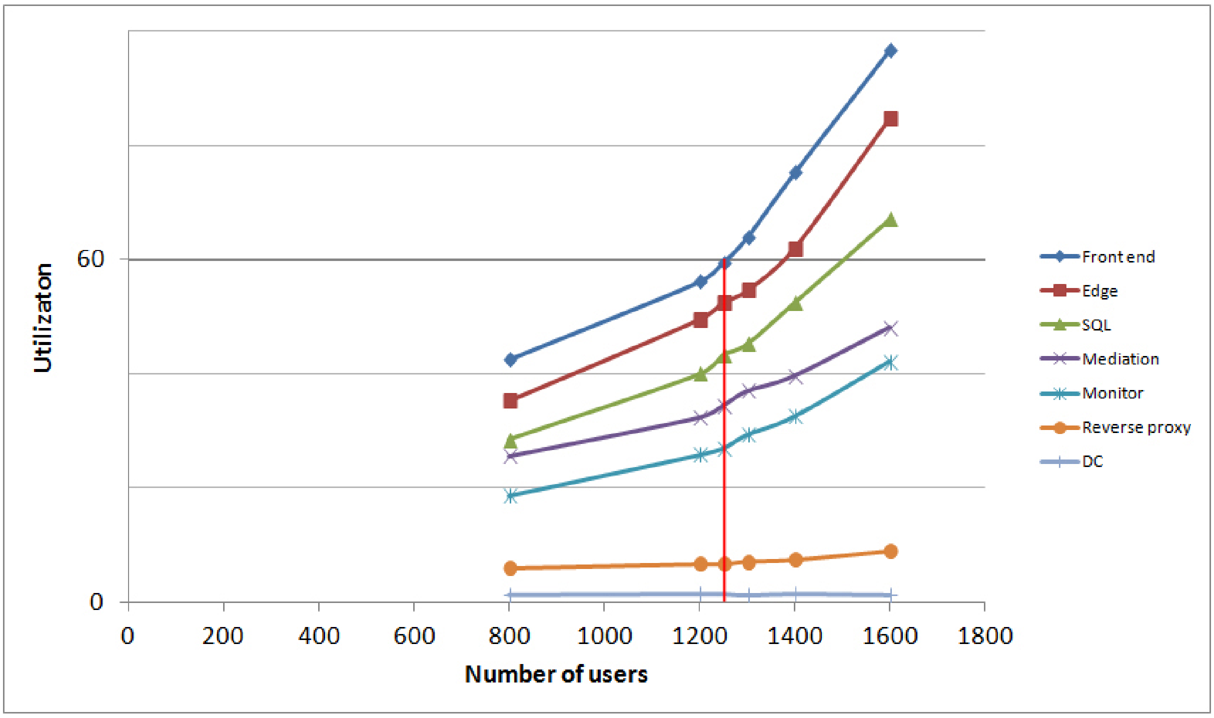

Figure 4.2: Normalized utilization of all servers, from the 4/4 system size high user load test

4.5 Scalability in Lync

Summarizing the results from table 4.1 gives the scalability two scalability curves for high and low user load shown in figure 4.3.

The results shows a close-to-linear scalability. There are several errors in the maximum number of users values and they should be considered estimates:

1. The binary search stops at a given number of iteration. This means that the unexplored search space i still large. Finding the "correct" value would require even more iterations, but due to time constraints this was not feasible.

2. Random measurement errors affects each test. This effect could be reduced by performing the same experiment several times and averaging the results. This would however require a lot more time.

3. Only the CPU resources are scaled, while memory, network and storage remain the same for all system sizes. Even though the CPU has proved to be the bottleneck device, the improper scaling of other resources could create a bias towards higher number of users for 1/4 and 2/4 system size.

4. The L3 cache in the 1/4 system size is scaled to 1/2 instead of 1/4. This may give a bias towards higher number of users because the processor can serve a relatively higher load before experiencing contention because of the increased cache size. This is especially relevant since the processor is shown to be the bottleneck device of this Lync deployment.

The first error in the above list can be reduced by using interpolation techniques. Sec-

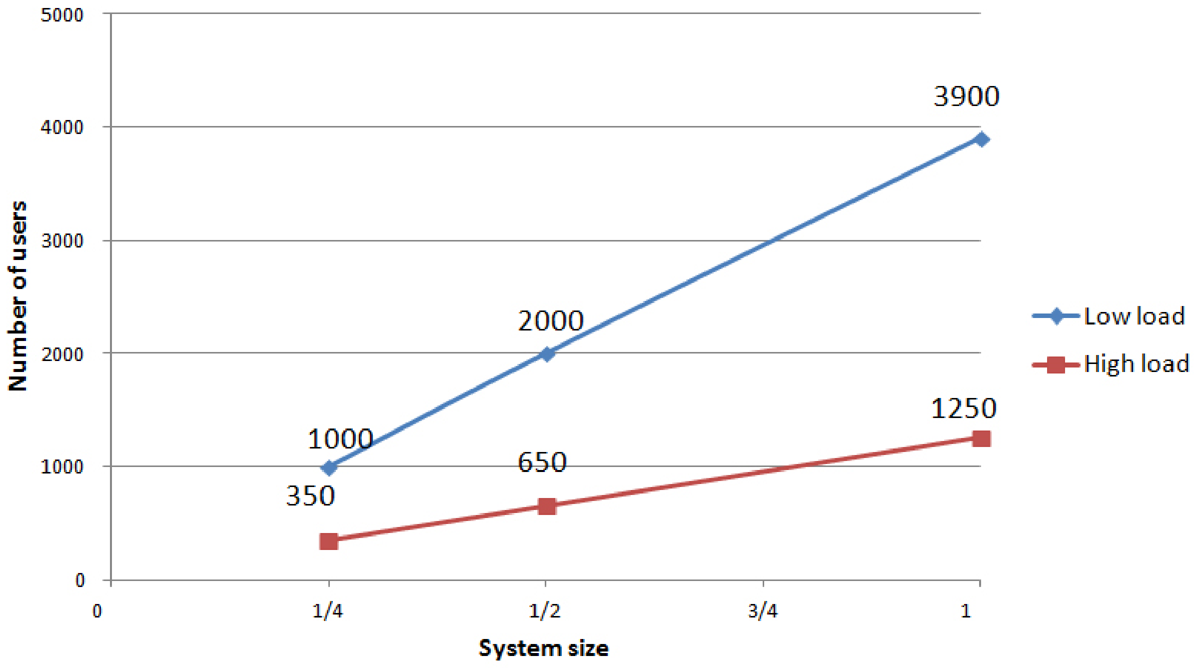

Figure 4.3: Measured scalability in Microsoft Lync

tion 3.3.1 describes why a binary search is preferred before an interpolation method when finding the maximum number of users. The quality parameters has no guarantee of linearity. It is however safer to assume linearity over a smaller range, like the remaining search space after using the binary search for several iteration. This technique gives a more precise estimate for the maximum number of users. However it is important to note that the MOS score is given in two digits only, so the round off error must be considered. The interpolation technique for finding the interpolated number of users ($N_{0.5}$) as well as lower and upper round off counters ($N_{0.5}^{-}$, $N_{0.5}^{+}$) is shown below

$$N_{0.5} = N_b + (N_a - N_b)\frac{(0.5 - MOS_b)}{(MOS_a - MOS_b)} \quad (4.1)$$

$$N_{0.5}^{-} = N_b + (N_a - N_b)\frac{(0.5 - MOS_b - 0.005)}{(MOS_a - MOS_b + 0.01)} \quad (4.2)$$

$$N_{0.5}^{+} = N_b + (N_a - N_b)\frac{(0.5 - MOS_b + 0.005)}{(MOS_a - MOS_b - 0.01)} \quad (4.3)$$

N_b and MOS_b denotes the values measured *below* the MOS threshold. N_a and MOS_a are *above* the threshold.

The interpolation results are summarized in table 4.2 and also shown in figure 4.4. The figure shows the $N_{0.5}$ values only. The error bars ($N_{0.5}^{-}$ and $N_{0.5}^{+}$) are not included in the figure because they were too small to view.

The interpolated values for number of users still show close-to-linear scalability. There is however a slightly sub-linear tendency in the results. This tendency still exists when the round off error is considered. The improper scaling of resources (as explained in point 3 and 4 on page 41) should both give a systematic bias towards higher number

Table 4.2: Scalability results with linear interpolation

User load	High			Low		
System size	1/4	2/4	4/4	1/4	2/4	4/4
N_b	350	650	1250	1000	2000	3900
N_a	400	700	1300	1100	2200	4050
MOS_b	0.5	0.45	0.44	0.43	0.47	0.47
MOS_a	0.59	0.64	0.55	0.52	0.59	0.52
$N_{0.5}^-$	348	661	1273	1065	2038	3963
$N_{0.5}$	350	663	1277	1078	2050	3990
$N_{0.5}^+$	353	665	1283	1094	2064	4031

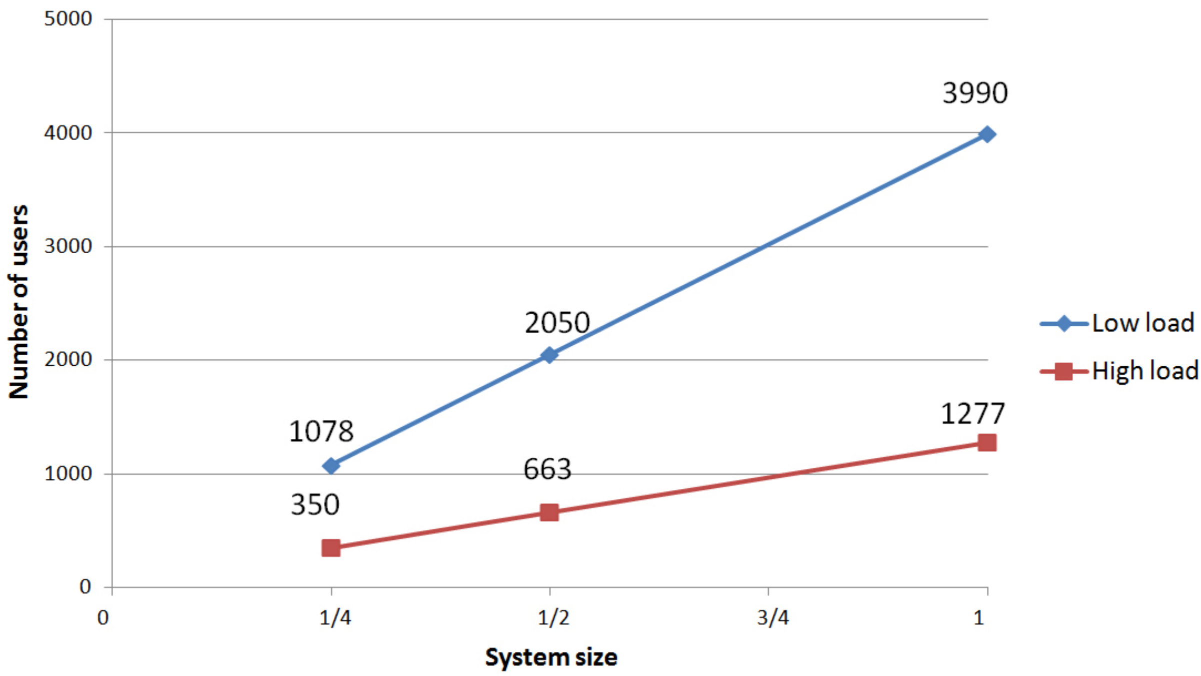

Figure 4.4: Interpolated scalability values

of users for 1/4 and 2/4 system size. However, it is hard to determine whether this bias alone accounts for the sub-linearity or whether the system really has a sub-linear scaling. Answering this question would require a different test setup where the bias was reduced or removed. An alternative test setup would be to run multiple identical Lync instances on the ESXi host instead of trying to scale down the system resources. This means that for the 1/4 system size scenario, a total of 4 identical Lync instances with the same load would run at the same time. Then the available resources for each Lync instance would be 1/4. However, there would be more contention on resources. This kind of test setup with multiple Lync instances could not be tested in this master thesis because installing Lync instances was out sourced from Telenor. New Lync instances would have to be ordered and the resulting delay would postpone the thesis work.

For practical concerns, such as when new Lync deployments are provisioned, the scalability in the 1/4 - 4/4 system size range can be considered linear. The sub-linear tendency that may exist in this system size range is too low to make an impact.

Chapter 5

Discussion

5.1 Scalability outside the measured system size range

The drawback of using empirical scalability tests rather than a full scalability model is that the results can only be reliable within the system size range that was actually measured. In this thesis the range is $1/4$ - $4/4$ physical servers. Using the Cisco UCS platform it is quite straightforward to increase the number of blade servers and include them in the same virtual resource pool. This would allow the system size to scale to e.g. $16/4$ by including 4 blade servers. There is however one constraint in the system setup that makes such configurations uninteresting. The standard edition license doesn't allow several front end servers to be allocated to the same front end pool. Only single front end servers can be set up. In order to set up several servers in a pool, an enterprise edition license is needed. With the given constraint, the best configuration to achieve high performance is to put the front end server VM on a single physical server and allocate 24 vCPUs for it. Then the remaining VMs can be put on the other blade servers. However the front end server VM requires more compute resources than the other VMs altogether. The results from table B.5 shows that the front end server has a % RUN value of 832, while the other servers sums to 600. This means that efficient provisioning can be achieved by putting the front end server on one physical server and the remaining servers on another physical server. Using more than two physical servers would not give significantly better performance, because the front end VM would have a much higher utilization and become a bottleneck.

A static performance model could give scalability predicates beyond 4/4 system size, but since the considered Lync instance cannot practically scale beyond 8/4 system size, the results gathered using empirical test cover half of the whole practical scalability range.

The Lync sizing study [19] shows that Lync scales linearly when the system size is varied from one to four front end servers. This sizing study uses the enterprise edition licence and has a different user scenario than this report. However, the linear scalability conclusion from the sizing study supports the belief that the sub-linear scaling tendency observed in this report is actually caused by measurement bias alone as discussed in section 4.5.

5.2 Time considerations

In order to fully understand the impact of a proposed research method it is important to also consider the amount of work required to use the method. As shown in appendix C, a lot of time was required to set up the test system, tune the Lync configuration to work with LyncPerfTool, resolve issues with the performance measure tools, etc. One can argue that this work would require less time for a person experienced with the specific technology who had direct access to the equipment instead of working remotely. However, the time required to perform the actual tests, would not have been affected by such concerns. The binary searches in this report were carried out in a total of 12 days. The work required for one iteration of the binary search is outlined in table 5.1.

Table 5.1: Time required for one binary search iteration

Approx time	Task
5 min	Create user profile configuration file
5 min	Prepare parameter capture scripts on ESXi hosts
5 min	Copy configuration file to all 3 LyncPerfTool clients and start the LyncPerfTool clients
90 min	Run test (including 30 min transient and 60 min measurement period)
10 min	Copy csv log file from ESXi hosts to home computer
5 min	Read QoE parameters from the monitoring server
10 min	Parse the results from the csv file
5 min	Compare parameters to thresholds and determine the correct direction for the next iteration of the binary search

The table shows that one iteration can be carried out in $\approx$ 2 hours 15 min. Configuring the tests and interpreting the results takes $\approx$ 45 min, half the time of actually running the test. This does of course require all steps to be carried out correctly. Since LyncPerfTool clients were situated both outside and inside the firewall (see figure 3.1), an extra network interface were added to give external user simulators direct access to the internal network. This made file sharing possible without altering the standard reverse proxy configuration. This interface was for pre-test file sharing only. If the extra network interface were not disabled before the test was started, all sessions that were set up using ICE would short-cut the edge server and use the extra network interface instead. Then the performance results were biased towards better performance estimates and the whole iteration had to be run all over again with a disabled network interface. This happened twice during the test period and affected the total test time.

Automating the parameter capture could save a lot of manual work. It is possible to write scripts to start LyncPerfTool and to automate almost all parts of the capture process. But there were one major and two minor issues that impede full automation. The major issue is that csv log files had to be copied from ESXi hosts using a Citrix Reciever GUI. Seemingly, there were no CLI options available. The minor issues were; (1) creating new configuration files from UserProfileGenerator without using the GUI and (2) reading monitoring server results from a web interface. These issues could probably be overcome.

A person with experience from Lync setup can configure a Lync environment in two-three days. Two-three more days are required to install the LyncPerfTool and configure the Lync scenario accordingly. Applying the same method as described in this report, the results can be reproduced in 16-18 days. If a different management setup were used (excluding remote access through Citrix Receiver) the parameter capture could probably be automated and the binary seaches could be fully automated.

Chapter 6

Conclusion

This master thesis has studied the scalability of a virtualized Microsoft Lync deployment. The servers where deployed on a Cisco UCS platform with one blade server. The results shows that there is a close-to-linear scaling from 1/4 to 4/4 system size. The scalability is comparable when considering both low and high user load. A new deployment should be scaled according to the number of simultaneous end-users.

When the Lync deployment is installed on a Cisco UCS platform, the CPU resources become the dominating bottleneck. The hardware configuration of Cisco UCS is by far comparable to hardware from other vendors as well. This means that upgrading to more CPU resources is most important in order to increase system capacity.

Mean opinion score is the best metric to monitor reduced service quality for end users. The results shows that this metric is the first to exceed thresholds. Another good performance measure is to monitor the utilization on the front end server. When the utilization exceeds 60%, service quality is degraded.

The results from this thesis will help Telenor as well as other users to provision their Lync deployments with the right amount of hardware resources. The provisioning of resources is dependent on the average user load, and this thesis suggests upper and lower counters based on Microsofts user load model.

Further work

From the findings in this master thesis, there are several interesting problems that needs further study. These problems can be a starting point for a new master thesis or other scientific studies:

- Explore the scalability of Microsoft Lync using an enterprise edition deployment that allows several servers in the same server pool. This would allow scaling out, instead of scaling up, as shown in this thesis.
- Tune the configuration of a Cisco UCS platform in order to find the most cost-effective hardware platform for hosting a virtual Lync deployment.

- Compare the performance of different hypervisors and find the optimal hypervisor for a virtualized Lync deployment by comparing the VMware hypervisor to other major hypervisors, like Microsoft Hyper-V.

Bibliography

[1] Mean opinion scores and metrics. http://technet.microsoft.com/en-us/library/bb894481(v=office.12).aspx, Jul 2010. Last visited: 2012 Jul 8th.

[2] Lync server 2010 user models. http://technet.microsoft.com/en-us/library/gg398811.aspx, Jul 2011. Last visited: 2012 Jul 8th.

[3] Lync stress and performance tool. http://www.microsoft.com/download/en/details.aspx?id=25005, Mar 2011. Last visited: 2012 Jul 8th.

[4] Magic quadrant for unified communications. http://www.gartner.com/technology/reprints.do?id=1-1728DMD\&ct=110823\&st=sb, Aug 2011. Last visited: 2012 Jul 8th.

[5] Media quality summary report. http://technet.microsoft.com/en-us/library/gg615012, Jul 2011. Last visited: 2012 Jul 8th.

[6] Microsoft lync server 2010: Lync edge server security. http://technet.microsoft.com/en-us/magazine/hh272676.aspx, Jun 2011. Last visited: 2012 Jul 8th.

[7] Reference topology with high availability and a single data center. http://technet.microsoft.com/en-us/library/gg425939, Mar 2011. Last visited: 2012 Jul 8th.

[8] Unified computing technology. http://www.cisco.com/en/US/products/ps10265/technology.html, Feb 2012. Last visited: 2012 Jul 8th.

[9] User and client authentication for lync server 2010. http://technet.microsoft.com/en-us/library/gg195618.aspx, Mar 2012. Last visited: 2012 Jul 8th.

[10] G. Millstein G. Brataas. Energy and deployment efficiency in telenor data centers. Jul 2011.

[11] Pacyk Lewis, Abbate. *Microsoft Lync Server 2010 Unleashed*. SAMS, 1st edition, 2011.

[12] Microsoft, http://www.microsoft.com/download/en/confirmation.aspx?id=6412. *Quality of Experience*, Oct 2007. Last visited: 2012 Jul 8th.

[13] Microsoft, http://www.microsoft.com/download/en/details.aspx?id=21646. *Microsoft Lync Server 2010 Planning Guide*, Aug 2011. Last visited: 2012 Jul 8th.

[14] Microsoft, http://technet.microsoft.com/en-us/library/gg398835. *Server Hardware Platforms*, Mar 2011. Last visited: 2012 Jul 8th.

[15] Microsoft, http://www.microsoft.com/download/en/details.aspx?id=12295. *Microsoft Lync Server 2010 Capacity Calculator*, Feb 2012. Last visited: 2012 Jul 8th.

[16] Microsoft, http://msdn.microsoft.com/en-us/library/cc431495(v=office.12).aspx. *[MS-ICE]: Interactive Connectivity Establishment (ICE) Extensions*, 4.0 edition, Apr 2012. Last visited: 2012 Jul 8th.

[17] J.S. Lovstad P.H. Hughes. A generic model for quantifiable software deployment. In *International Conference on Software Engineering Advances*, pages 22–31, Aug 2007.

[18] P. Seeling. Network performance evaluation of microsoft office communications server 2007. In *IEEE International conference on Electro/Information Technology (EIT), 2011*, pages 1–6, May 2011.

[19] Global Engineering Solutions. Sizing study: Virtualized microsoft lync server 2010 deployment. Apr 2012.

[20] International Telecommunication Union. Methods for subjective determination of transmission quality. Aug 1996.

[21] VMware, http://www.vmware.com/files/pdf/hpm-perf-vsphere5.pdf. *Host Power Management in VMware vSphere 5*, Aug 2011. Last visited: 2012 Jul 8th.

[22] VMware, http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-resource-management-guide.pdf. *vSphere Resource Management Guide, Update 1 ESXi 5.0*, 2012. Last visited: 2012 Jul 8th.

[23] Hanna Winters. *Mastering Microsoft Lync Server 2010*. Sybex, 1st edition, 2012.

[24] Jin Li Philip A. Chou Y. Richard Yang Ye Wang, Cheng Huang. Qosaas: Quality of service as a service. In *11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services*, pages 1–6, Mar 2011.

Appendix A

LyncPerfTool XML configuration script

DESCRIPTION

This script fixes a bug in the UserProfileGenerator.exe program used to generate XML files for the LyncPerfTool.exe program. LyncPerfTool uses a parameter called *SipAuthMethod* to decide which of the authentication methods to use. There are four possible methods of authentication [6, 9]:

Kerberos	MIT Kerberos version 5 security protocol is the standard authentication protocol for all internal users with active directory credentials. Kerberos requires client connectivity to Active Directory Domain Services, which is why it cannot be used for authenticating clients outside the corporate firewall.
NTLM	NT LAN Manager is for users with Active Directory credentials who are connecting from an endpoint outside the corporate firewall. The access edge service passes logon requests to the front end server for authentication. The access edge service itself performs no authentication. NTLM is a weaker authentication protocol than Kerberos.
Digest	The digest protocol for so-called anonymous users. Anonymous users are outside users who do not have recognized Active Directory credentials but who have been invited to an on-premises conference and possess a valid conference key. Digest authentication is not used for other client interactions.
TLS-DSK	TLS-DSK is certificate-based authentication. Upon successfully signing in to Lync Server the first time, the Lync client requests a client certificate. Lync Server issues a self-signed certificate, which the Lync client uses for all subsequent sign-ins. This certificate is renewed every six months.

The “SipAuthMethod” is always set to “Kerberos” by UserProfileGenerator.exe. If this is not manually changed to “NTLM”, LyncPerfTool will not be able to sign in external users.

CODE LISTING

```
foreach ($file in get-childitem .\*.* -include *.xml {
        (Get-Content $file) | `
        ForEach-Object { $_ -replace ''Kerberos'' ''NTLM'' } | `
        Set-Content $file -Encoding Unicode
}
```

USAGE

1. Copy script file to directory containing xml configuration files.
2. Make sure script file extension is “ps1” as a valid PowerShell script
3. Right click and select “Run with PowerShell”

EXPLANATION

The script iterates through all files with “.xml” file extensions in the current directory. Each file is rewritten and any occurance of the phrase “Kerberos” is replaced by “NTLM”.

REQUIREMENTS

Needs PowerShell 1.x or higher (PowerShell 2.0 is installed by default on Windows Server 2008 R2).

Appendix B

Performance results from binary searches

Table B.1: Binary search for high load on 1/4 system size

Number of users	200	300	350	400
Lync server esxtop statistics: Group CPU				
% RUN front end	136,2	192,908	232,3	296,108
% RUN edge	56,24	85,961	103,696	125,126
% RUN SQL	24,312	33,25	38,353	47,946
% RUN mediation	12,54	15,834	18,797	21,484
% RUN monitoring	9,243	10,417	11,827	13,171
% RUN rev. proxy	14,251	4,534	4,777	5,112
% RUN domain contr.	1,052	1,089	1,034	0,985
Lync server esxtop statistics: Memory				
Swap Mbytes Read/sec	0	0	0	0
Swap Mbytes Read/sec	0	0	0	0
Lync server esxtop statistics: Network				
Mbit Rec/sec mediation	0,52	0,771	0,878	1,026
Mbit Tra/sec mediation	0,491	0,752	0,832	0,978
Mbit Rec/sec edge	7,821	11,416	13,212	15,23
Mbit Tra/sec edge	11,612	17,412	220,351	23,148
Mbit Rec/sec rev.proxy	0,006	0,007	0,007	0,009
Mbit Tra/sec rev.proxy	0,007	0,012	0,014	0,014
Lync server esxtop statistics: Physical disk				
Average driver ms/comm	5,013	5,034	5,068	5,088
Average kernel ms/comm	0,015	0,015	0,015	0,015
Average guest ms/comm	5,038	5,059	5,093	5,113
LyncPerfTool esxtop statistics: Group CPU				
% RUN client 1	576,32	683,804	848,112	946,941
% RUN client 2	572,241	690,127	850,621	941,319
% RUN PSTN gateway	8,652	10,972	12,684	14,321
LyncPerfTool performance counters				
Local 503 responses/sec	0	0	0	0.01
Local 504 responses/sec	0	0	0	0
Monitoring server statistics				
Round trip time UC conf	4	9	39	46
Jitter UC conf	2	4	13	14
Packet loss UC conf	0	0	0,05	0,06
MOS degradation UC conf	0,04	0,28	0,41	0,63
Round trip time UC leg	3	8	32	40
Jitter UC leg	2	4	13	16
Packet loss UC leg	0	0	0,06	0,07
MOS degradation UC leg	0,07	0,15	0,5	0,59
Round trip time GW leg	2	4	5	7
Jitter GW leg	1	2	4	5
Packet loss GW leg	0	0	0	0
MOS degradation GW leg	0,1	0,14	0,19	0,21

Table B.2: Binary search for low load on 1/4 system size

Number of users	800	1000	1100	1200	1600
Lync server esxtop statistics: Group CPU					
% RUN front end	192,804	235,612	269,404	314,580	397,404
% RUN edge	84,724	104,864	114,582	130,436	179,484
% RUN SQL	28,197	34,29	37,719	41,148	54,864
% RUN mediation	12,85	15,925	17,431	19,128	26,491
% RUN monitoring	9,351	11,091	12,302	13,561	18,45
% RUN rev. proxy	3,613	4,589	5,261	5,631	7,451
% RUN domain contr.	0,994	1,091	1,2001	1,309	1,7456
Lync server esxtop statistics: Memory					
Swap Mbytes Read/sec	0	0	0	0	0
Swap Mbytes Read/sec	0	0	0	0	0
Lync server esxtop statistics: Network					
Mbit Rec/sec mediation	1,384	1,787	1,954	2,192	2,869
Mbit Tra/sec mediation	1,377	1,760	1,898	1,997	2,701
Mbit Rec/sec edge	21,201	26,668	29,162	32,135	42,273
Mbit Tra/sec edge	32,192	40,341	44,145	48,135	64,001
Mbit Rec/sec rev.proxy	0,012	0,016	0,172	0,019	0,026
Mbit Tra/sec rev.proxy	0,019	0,024	0,027	0,029	0,038
Lync server esxtop statistics: Physical disk					
Average driver ms/comm	5,063	5,071	5,081	5,094	5,103
Average kernel ms/comm	0,015	0,015	0,015	0,015	0,015
Average guest ms/comm	5,088	5,096	5,106	5,119	5,128
LyncPerfTool esxtop statistics: Group CPU					
% RUN client 1	765,133	961,132	1065,432	1164,643	1558,169
% RUN client 2	770,416	965,183	1078,516	1169,131	1557,161
% RUN PSTN gateway	13,748	17,185	18,904	20,622	27,496
LyncPerfTool performance counters					
Local 503 responses/sec	0	0	0	0,01	0,03
Local 504 responses/sec	0	0	0	0	0
Monitoring server statistics					
Round trip time UC conf	16	23	36	39	43
Jitter UC conf	4	5	7	12	13
Packet loss UC conf	0	0,01	0,02	0,04	0,09
MOS degradation UC conf	0,21	0,43	0,51	0,56	0,67
Round trip time UC leg	14	20	32	34	39
Jitter UC leg	2	4	6	10	12
Packet loss UC leg	0	0	0,01	0,04	0,07
MOS degradation UC leg	0,19	0,41	0,52	0,59	0,69
Round trip time GW leg	2	3	4	6	9
Jitter GW leg	1	1	2	3	4
Packet loss GW leg	0	0	0	0	0
MOS degradation GW leg	0,13	0,19	0,21	0,23	0,28

Table B.3: Binary search for high load on 2/4 system size

Number of users	400	600	650	700	800
Lync server esxtop statistics: Group CPU					
% RUN front end	257,53	373,268	412,822	453,852	584,637
% RUN edge	126,33	174,393	200,972	222,903	277,771
% RUN SQL	53,26	68,056	76,253	83,767	94,424
% RUN mediation	22,23	32,447	35,383	38,756	42,627
% RUN monitoring	14,51	17,655	18,608	19,094	21,144
% RUN rev. proxy	5,13	5,467	5,706	5,813	5,93
% RUN domain contr.	1,193	1,298	1,386	1,405	1,253
Lync server esxtop statistics: Memory					
Swap Mbytes Read/sec	0	0	0	0	0
Swap Mbytes Read/sec	0	0	0	0	0
Lync server esxtop statistics: Network					
Mbit Rec/sec mediation	1,023	1,642	1,834	1,869	2,054
Mbit Tra/sec mediation	0,913	1,465	1,53	1,643	1,892
Mbit Rec/sec edge	14,864	21,913	23,689	25,613	29,15
Mbit Tra/sec edge	22,156	32,581	35,129	38,945	43,932
Mbit Rec/sec rev.proxy	0,009	0,015	0,015	0,017	0,019
Mbit Tra/sec rev.proxy	0,013	0,019	0,022	0,022	0,026
Lync server esxtop statistics: Physical disk					
Average driver ms/comm	5,101	5,092	5,089	5,098	5,132
Average kernel ms/comm	0,015	0,015	0,015	0,015	0,015
Average guest ms/comm	5,126	5,117	5,114	5,123	5,157
LyncPerfTool esxtop statistics: Group CPU					
% RUN client 1	929,391	1171,024	1259,059	1301,005	1483,727
% RUN client 2	934,12	1169,109	1262,091	1303,01	1481,462
% RUN PSTN gateway	14,265	17,568	19,196	19,662	21,541
LyncPerfTool performance counters					
Local 503 responses/sec	0	0	0	0	0,02
Local 504 responses/sec	0	0	0	0	0
Monitoring server statistics					
Round trip time UC conf	4	14	42	49	52
Jitter UC conf	1	6	6	13	17
Packet loss UC conf	0	0	0,03	0,06	0,08
MOS degradation UC conf	0,03	0,07	0,45	0,64	0,78
Round trip time UC leg	3	7	13	17	18
Jitter UC leg	1	3	7	7	7
Packet loss UC leg	0	0	0,04	0,05	0,05
MOS degradation UC leg	0,06	0,13	0,23	0,31	0,34
Round trip time GW leg	2	2	3	3	3
Jitter GW leg	1	1	1	2	2
Packet loss GW leg	0	0	0	0	0
MOS degradation GW leg	0,05	0,11	0,12	0,11	0,12

Table B.4: Binary search for low load on 2/4 system size

Number of users	1200	1800	2000	2200	2400
Lync server esxtop statistics: Group CPU					
% RUN front end	270,291	372,407	413,882	474,901	557,837
% RUN edge	129,672	183,676	202,572	222,44	251,256
% RUN SQL	48,262	70,522	76,53	86,432	93,832
% RUN mediation	22,45	32,846	35,915	40,265	43,816
% RUN monitoring	11,213	16,461	18,068	20,513	21,946
% RUN rev. proxy	3,516	5,42	5,756	6,512	6,987
% RUN domain contr.	1,203	1,168	1,294	1,361	1,281
Lync server esxtop statistics: Memory					
Swap Mbytes Read/sec	0	0	0	0	0
Swap Mbytes Read/sec	0	0	0	0	0
Lync server esxtop statistics: Network					
Mbit Rec/sec mediation	2,146	3,205	3,587	3,939	4,168
Mbit Tra/sec mediation	2,007	2,995	3,3	3,618	3,912
Mbit Rec/sec edge	30,400	45,647	50,67	56,411	60,915
Mbit Tra/sec edge	45,866	67,601	76,237	84,229	92,026
Mbit Rec/sec rev.proxy	0,019	0,03	0,033	0,036	0,04
Mbit Tra/sec rev.proxy	0,026	0,040	0,045	0,050	0,055
Lync server esxtop statistics: Physical disk					
Average driver ms/comm	5,076	5,084	5,091	5,126	5,143
Average kernel ms/comm	0,015	0,015	0,015	0,015	0,015
Average guest ms/comm	5,101	5,109	5,116	5,151	5,168
LyncPerfTool esxtop statistics: Group CPU					
% RUN client 1	864,583	1290,543	1436,543	1581,154	1753,217
% RUN client 2	875,563	1295,613	1451,654	1584,563	1742,65
% RUN PSTN gateway	15,846	23,192	26,841	28,516	31,195
LyncPerfTool performance counters					
Local 503 responses/sec	0	0	0	0,01	0,02
Local 504 responses/sec	0	0	0	0	0
Monitoring server statistics					
Round trip time UC conf	3	12	36	45	48
Jitter UC conf	1	5	7	12	18
Packet loss UC conf	0	0,02	0,04	0,06	0,08
MOS degradation UC conf	0,03	0,24	0,47	0,59	0,78
Round trip time UC leg	2	6	10	16	20
Jitter UC leg	1	2	4	5	7
Packet loss UC leg	0	0	0,03	0,06	0,07
MOS degradation UC leg	0,07	0,23	0,26	0,33	0,37
Round trip time GW leg	2	2	2	3	3
Jitter GW leg	1	1	1	1	2
Packet loss GW leg	0	0	0	0	0
MOS degradation GW leg	0,02	0,13	0,15	0,19	0,21

Table B.5: Binary search for high load on 4/4 system size

Number of users	800	1200	1250	1300	1400	1600
Lync server esxtop statistics: Group CPU						
% RUN front end	594,664	785,310	831,978	892,782	1051,092	1351,409
% RUN edge	246,428	346,038	367,038	381,43	433,463	592,203
% RUN SQL	85,257	119,517	129,519	135,156	156,913	200,73
% RUN mediation	50,964	64,484	68,684	73,73	79,14	96,04
% RUN monitoring	18,658	25,767	26,897	29,275	32,48	42,05
% RUN rev. proxy	5,926	6,64	6,65	7,08	7,41	8,91
% RUN domain contr.	1,283	1,427	1,427	1,184	1,412	1,24
Lync server esxtop statistics: Memory						
Swap Mbytes Read/sec	0	0		0	0	0
Swap Mbytes Read/sec	0	0		0	0	0
Lync server esxtop statistics: Network						
Mbit Rec/sec mediation	2,045	3,12	3,245	3,451	3,612	4,12
Mbit Tra/sec mediation	1,662	2,399	2,612	2,71	2,956	3,651
Mbit Rec/sec edge	28,192	42,31	44,621	46,32	50,023	56,431
Mbit Tra/sec edge	46,536	69,21	73,561	75,629	82,36	94,21
Mbit Rec/sec rev.proxy	0,018	0,031	0,032	0,032	0,033	0,036
Mbit Tra/sec rev.proxy	0,028	0,041	0,043	0,046	0,048	0,056
Lync server esxtop statistics: Physical disk						
Average driver ms/comm	5,512	5,622	5,724	5,805	5,657	5,888
Average kernel ms/comm	0,015	0,015	0,015	0,015	0,015	0,016
Average guest ms/comm	5,528	5,638	5,740	5,821	5,672	5,905
LyncPerfTool esxtop statistics: Group CPU						
% RUN client 1	1494,739	1709,264	1803,219	1865,213	2002,346	2086,409
% RUN client 2	1494,196	1713,646	1809,198	1861,849	2011,563	2094,385
% RUN PSTN gateway	21,757	26,11	27,369	28,241	28,799	31,135
LyncPerfTool performance counters						
Local 503 responses/sec	0	0	0	0	0,02	0,04
Local 504 responses/sec	0	0	0	0	0	0,01
Monitoring server statistics						
Round trip time UC conf	7	36	42	44	52	134
Jitter UC conf	2	6	7	9	11	23
Packet loss UC conf	0	0,03	0,04	0,06	0,09	0,12
MOS degradation UC conf	0,02	0,18	0,44	0,55	0,78	0,82
Round trip time UC leg	5	7	9	11	18	24
Jitter UC leg	2	4	4	5	8	13
Packet loss UC leg	0	0,02	0,04	0,05	0,08	0,08
MOS degradation UC leg	0,08	0,3	0,41	0,49	0,58	0,62
Round trip time GW leg	2	2	2	3	5	7
Jitter GW leg	1	1	2	2	3	3
Packet loss GW leg	0	0	0	0	0	0
MOS degradation GW leg	0,08	0,09	0,09	0,1	0,12	0,14

Table B.6: Binary search for low load on 4/4 system size

Number of users	2400	3600	3900	4050	4200	4800
Lync server esxtop statistics: Group CPU						
% RUN front end	523,992	770,854	835,145	883,68	959,182	1183,504
% RUN edge	230,615	342,412	368,291	381,717	417,291	520,933
% RUN SQL	80,748	120,645	129,843	135,639	140,739	161,538
% RUN mediation	43,062	64,286	69,184	71,884	74,922	85,632
% RUN monitoring	16,926	25,134	27,017	28,164	29,341	33,312
% RUN rev. proxy	4,561	6,181	6,691	6,981	7,213	8,241
% RUN domain contr.	1,316	1,254	1,364	1,405	1,384	1,435
Lync server esxtop statistics: Memory						
Swap Mbytes Read/sec	0	0	0	0	0	0
Swap Mbytes Read/sec	0	0	0	0	0	0
Lync server esxtop statistics: Network						
Mbit Rec/sec mediation	4,251	6,388	6,989	7,256	7,545	8,636
Mbit Tra/sec mediation	3,806	5,731	6,248	6,414	6,575	7,636
Mbit Rec/sec edge	58,811	88,011	95,514	99,194	103,720	116,823
Mbit Tra/sec edge	99,272	148,957	158,563	167,523	172,565	196,03
Mbit Rec/sec rev.proxy	0,036	0,058	0,061	0,062	0,066	0,0756
Mbit Tra/sec rev.proxy	0,058	0,087	0,094	0,098	0,103	0,121
Lync server esxtop statistics: Physical disk						
Average driver ms/comm	5,539	5,684	5,716	5,778	5,861	5,916
Average kernel ms/comm	0,015	0,015	0,015	0,015	0,016	0,016
Average guest ms/comm	5,555	5,900	5,832	5,894	5,878	5,933
LyncPerfTool esxtop statistics: Group CPU						
% RUN client 1	1266,306	1878,234	2050,364	2135,41	2215,943	2365,142
% RUN client 2	1289,34	1885,543	2053,647	2134,516	2217,463	2376,153
% RUN PSTN gateway	22,916	34,065	36,219	37,516	39,698	45,513
LyncPerfTool performance counters						
Local 503 responses/sec	0	0	0	0	0,01	0,03
Local 504 responses/sec	0	0	0	0	0	0,01
Monitoring server statistics						
Round trip time UC conf	2	14	25	34	46	105
Jitter UC conf	1	4	5	7	10	21
Packet loss UC conf	0	0,02	0,03	0,05	0,1	0,13
MOS degradation UC conf	0,03	0,19	0,47	0,52	0,69	0,76
Round trip time UC leg	3	5	8	10	16	36
Jitter UC leg	1	2	3	3	5	12
Packet loss UC leg	0	0,01	0,03	0,05	0,09	0,12
MOS degradation UC leg	0,07	0,26	0,38	0,46	0,56	0,68
Round trip time GW leg	2	2	3	4	6	9
Jitter GW leg	1	1	1	2	2	4
Packet loss GW leg	0	0	0	0	0	0
MOS degradation GW leg	0,09	0,12	0,13	0,16	0,19	0,26

Appendix C

Master thesis work log

This appendix contains a log describing the work done in this master thesis. Each record is followed by a date, which shows when the work was started. Such a log can prove useful for a reader who is trying to reproduce the results by using the same method. Many scientific studies describe solution methods that perform very well on a given problem, but they forget to mention how long it takes to carry out the method in terms of test setup, configuration, instrumentation, information gathering and processing.

Scientific projects tries to find better methods for achieving a certain task. In order to evaluate different proposed methods one should not only consider the results from using the method, but also the time it takes to use the method in terms of test setup, instrumentation, time for measurements, etc. In order to allow readers to evaluate this thesis, a detailed thesis log is included. This log also explains some of the small and large problems that were encountered and how they were solved or otherwise overcome. Each record in the log has a date stamp to show the amount of time used for each task.

Period	**Task**
18-Jan	Start of master thesis. Task description, formal agreements, etc.
20-Jan	Kick of telephone conference with all supervisors. Ambitions for the work, a preliminary architecture model, a coarse timeline describing the work process.
21-Jan	Studied reporting tools from BMC and Microsoft, read reports from Telenor/SINTEF collaboration
30-Jan	Physical meeting at Fornebu. Decided to specify the task to scalability measurements on Microsoft Lync. Focus on modelling Lync, modelling Vmware and modelling the hardware platform Cisco UCS
31-Jan →	Getting to know Lync as a system. Architecture, server roles, many different deployment options. Studying Vmware hypervisor technology. Resource allocation, overhead, etc Studying Cisco UCS hardware. How are the components linked together. How can this system be modelled?

8-Feb	Telephone conference with all supervisors: Generate a model using SP and queueing models. Using a test environment to evaluate the model. How to generate load for the Lync test.
16-Feb	Telephone conference with all supervisors: Ask for a detailed deployment scetch of the Lync system Telenor has started setting up the Cisco UCS platform
1-Mar	The UCS hardware platform is done. However, the installation of Lync is not complete. Measurements can start in week 13.
15-Mar	Telephone conference. Alexander and Georg presented as system administrator contacts. Request information on the Lync setup.
29-Mar	Telephone conference. Testing can start in week 15-16
13-Apr	Access to Telenor Remote Access is granted.
	Started to test TRA connection. Got to know the different tools
17-Apr	Resolved issue with missing permissions for remote sftp login.
18-Apr	Resolved issue with uncorrect directory mapping on winscp server. The error prevented any file transfer.
19-Apr	Telephone conference. Go/Nogo for project. The Lync instance will be setup by Dax in the next week.
19-Apr	A total review of tools required to perform the test. The following components are missing: • Access to Cisco UCS Manager to measure bandwidth usage. Preferrably both a GUI to get to know the system, but also a CLI to run automated scriptet measure operations. • Access to resxtop on each ESXi host. Need IP addresses and authentication information • Access to VMWare vCenter Operations Manager. This is required to customize resource allocations as well as managing VM snapshots. • A network map of all units in the management network. This will help resolving issues regarding network access, permissions, transferring files, etc. • Documentation on the running Lync instance. Server roles, IP addresses, resource allocations, etc. Much of this could probably be read from vCenter • "Jumphost" (the machine used for LyncPerfTool) must be in a separate resource group. This is beacuse the the Lync server performance measures should not be affected by the LyncPerfTool process. • A monitor session on the fabric interconnect that copies all network traffic to "jumphost" would be of great help when troubleshooting internal Lync errors. Jumphost should have a dedicated virtual network interface to recieve this traffic."
20-Apr	Got access to UCS Manager through TRA, but the remote application failed to start.
21-Apr	Successfully mapped the directory for sftp access on "jumphost". I got the updated correct path from Georg.

21-Apr	With a working sftp connection I started configuring the jumphost server and installing the LyncPerfTool software
23-Apr	The issue with the UCS Manager application that failed to start was fixed by installing Mozilla Firefox instead of Internet Explorer
23-Apr	Requested admin password for the Lync servers
24-Apr	Got domain administrator password for the Lync domain. I'll have to wait untill Datametrix has finished installing certificates and completing the Lync installation before logging into the servers
24-Apr	SSH server and esxi console was enabled for all ESXi hosts. This was done by running a KVM console on each machine from UCS Manager.
24-Apr	I discovered that LyncPerfTool required at least to separate VMs for external client simulation as well as PSTN gateway simulation. These needed different network locations. In addition these servers had to be members of the same domain as the Lync servers. This means that "jumphost" cannot be used to run LyncPerfTool.
25-Apr	Two new VMs were created for running LyncPerfTool. The first VM was connected to both Vlan 1030 (the external sub network of the DMZ) as well as Vlan 2030 (the internal sub network of the DMZ). The second VM was connected to Vlan 2031 (the Lync internal server sub network) only.
26-Apr	Started preparing the new servers and installing LyncPerfTool all over again.
26-Apr	Recieved a notice that a monitor session from the fabric interconnect could not be copied to a virtual interface. This meant that in order to read the network traffic copy, a physical server would be needed.
30-Apr	A separate physical server (UCSSNIFFER) was set up to recieve traffic from the monitor session. This is accessible through remote desktop. The monitor session is however not enabled yet. This must be done by the network group.
3-May	Started running configuration scripts for LyncPerfTool. As a part of the process Lync dummy users where created. In the configuration process I noticed that I recieved errors on the PSTN gateway configuration. I also had problems creating dummy contact lists on the SQL server.
3-May	I noticed that there were no suitable way of exporting csv log files from the ESXi hosts. I had tried connecting to the ssh server from "jumphost", but this was not possible since the ESXi hosts were located in an isolated, high security management network that blocked incoming requests.
4-May	UCSSNIFFER is ready to recieve network traffic. I started installing wireshark for network capture. The server recieved a lot of traffic and everything seemed to be okay.
4-May	My vCenter profile was granted access to transfer files directly from an ESXi host to the Citrix client computer. This resolved the issue with copying csv log files from esxtop.

10-May	Some remaining issues regarding configuration of LyncPerfTool remains. Since noone in Telenor has first hand experience with this tool, I try to find answers to my questions by contacting Microsoft Norway.
15-May	File transfers from the ESXi hosts were suddenly denied. This seemed to be a client side error, but even reinstalling Citrix Reciever and the Citrix plugin software didn't resolve the issue.
21-May	Remote users cannot has no domain name to connect to the outside interface of the edge server or reverse proxy. This was fixed by adding new DNS A records to the central DNS server at the domain controller.
21-May	The error with file transfers from ESXi hosts was resolved. This error was probably due to a software update, where the new version of Citrix Reciever prohibits file system access for all new connections by default. This was solved by manually configuring Citrix Reciever for every new connection: Citrix Reciever-¿Preferences-¿Plug-in status-¿Connection Center-¿Session Security-¿Files-¿ Change setting from "Read Only" to "Full Access".
22-May	"By looking at the deployment overview found at the front end server I discovered that there were several differences between the actual configuration and the Lync deployment information I had received from Telenor. No monitoring server or mediation server were assiated in the deployment. There was a windows server called t-montst and another called t-medtst, but they both it seemed to have only a default Windows installaton The front end server was used a management repository even though there were a separate VM called t-mantst, supposed to be a managament server The front end server has no assosiated SQL store, this means that there is no separate back end server These problems were communicated to Telenor"
23-May	"By looking at the deployment overview found at the front end server I discovered that there were several differences between the actual configuration and the Lync deployment information I had received from Telenor. No monitoring server or mediation server were assiated in the deployment. There was a windows server called t-montst and another called t-medtst, but they both it seemed to have only a default Windows installaton The front end server was used a management repository even though there were a separate VM called t-mantst, supposed to be a managament server The front end server has no assosiated SQL store, this means that there is no separate back end server The reverse proxy doesn't seem to be configured and is neither a member of the Lync domain These problems were communicated to Telenor"
24-May	The strange Lync configuration was unexpected by Telenor who had ordered a regular hosted Lync installation from Datametrix. I was encouraged to contact the consultant in Datametrix to clear any misunderstanding and order a proper system reconfiguration.

24-May	I talked to the consultant in Datametrix. He told me that the missing configuration of monitoring server and mediation server was because no PSTN gateway was configured for the depoyment and he therefore concidered these server roles unneccesary. I explained that LyncPerfTool would act as a PSTN GW. The management server was not used in this deployment. The front end server uses a standard edition lisence. This means that the back end server role is integrated on the front end server. The monitoring server however has a separate server for SQL backend to increase performance. The reverse proxy server was installed and functioning, but it is not (and should not be) a member of the active directory domain. We agreed that the missing server roles would be installed, and the LyncPerfTool client would be assosiated as a PSTN gateway. He recommended that the mediation server and PSTN gateway used a secure protocol for communication, but I was unsure whether LyncPerfTool supported this. He would generate the required server certificate and install it on the LyncPerfTool server.
28-May	I discovered that LyncPerfTool had problems signing in external users.

31-May	I finally resolved the issue with remote user login. Posting my problem to technet didn't give any results, but after I figured out the solution on my own, I posted the answer in the same thread so that other users could benefit from it. http://social.technet.microsoft.com/Forums/da-DK/ocscapacityplanning/thread/b480d281-1dc6-4104-b4ae-ad01b5b09b10. The issue was solved by studying network traffic to/from the edge server. Close inspection of SIP message bodies showed that the clients tried to sign in using the Kerberos protocol. However, the Kerberos server was not accessible to remote users. After reading the documentation on Lync I found that remote users can use another protocol called NTLM to sign in when Kerberos server is inaccessible. I didn't know if LyncPerfTool supported the NTLM protocol or how this was configured. I started looking at the LyncPerfTool xml configuration files generated by the UserProfileGenerator program. There was an xml attribute called "SipAuthMethod" and this was always set to "Kerberos". I tried changing this to "NTLM" and watched the network traffic. Users were now able to setup an encrypted SIP session and exchanged several packets, but they were still unable to log in. By accessing the SIP error logs on the front end server while running LyncPerfTool I discovered that users that tried to sign in got rejected because their passwords didn't match. The very user accounts and passwords could be signed in as internal users by the Kerberos protocol. I suspected that this might be due to a erroneous signed character handling in the NTLM protocol implementation. I therefore created a new set of users with user prefixes containing small case letters only. When these users tried to sign in externally using the NTLM protocol, the authentication matched and the sign in was successful. To avoid manually changing the "SipAuthMethod" parameter of every configuration file, I created a custom script for this task. This script is documented in appendix A.
1-Jun	All ESXi hosts are set to high performance power management policy.
2-Jun	The mediation server is unable to establish a connection to the mediation server and vica verca. I'm trying to look at the network traffic to diagnose the connection problems.
2-Jun	During network troubleshooting on the PSTN gateway traffic it became clear that UCSSNIFFER did not receive all network traffic.
4-Jun	The problem with the monitoring session was due to a redundant network connection to the fabric interconnect. The monitoring session can only monitor one of the connection. The problem will be fixed by removing the redundant connection and forcing all traffic on one link, but removing redundancy is not desirable for the operational Telenor customer servers running in the same server chassis. The monitor session problem was fixed by migrating all servers in Tst-servers and Tst-clients to a separate server chassis without redundant network connection. After this change UCSSNIFFER succesfully received all network traffic.

5-Jun	After posting the encrypted PSTN GW connection problem on Technet I received a reply from a Microsoft forum operator that this functionality was probably not supported by LyncPerfTool. http://social.technet.microsoft.com/Forums/da-DK/ocscapacityplanning/thread/00dd1a2e-18b7-4cb4-b368-eaff2b67c38d
6-Jun	I discovered that it was not possible to allocate more than 8 vCPUs to the VM running LyncPerfTool for external users. This would reduce the performance of LyncPerfTool to 1/3 and may not be enough for load generating on the 4/4 system size.
7-Jun	Two new LyncPerfTool clients was added and provisioned with 24 vCPU. Now LyncPerfTool could run in parallel on two servers, generating sufficient load for testing.
7-Jun	Installed LyncPerfTool (again) on the two new servers and started systematic testing.
17-Jun	I was unable to authenticate to the TRA connection. I had almost finished the testing, but now it was suddenly disrupted. I contacted Telenor and explained the issue.
18-Jun	There had been a certificate error on the TRA system that prevented OTP to be sent. I was not the only TRA user affected. The issue was fixed now.
19-Jun	Completed the last performance tests.
21-Jun	Presented the results from my master thesis on a Telenor meeting at Fornebu.
22-Jun →	Completing the master thesis report.