NTNU

Norwegian University of Science and Technology

Optimal resource allocation in private clouds

TTM4511: Networks and Quality of Service

Specialization Project

Amir Afroozeh

Supervised By: Andres Gonzalez, Humberto Castejon
Professor: Bjarne E. Helvik

Trondheim, Norway
December 2012

Acknowledgements

I would like to thank my supervisors Andres Gonzalez and Humberto Castejon for their patience and valuable guidance during the process of completing this specialization project. Moreover, I would like to express my appreciation to Bjarne E. Helvik for helping me during the project.

Abstract

This report is written in order to highlight the work performed during the specialization project. The project is about optimal resource allocation in private clouds. The report contains three different parts. The first part is studying the state of the art of resource allocation in cloud computing. It continues with the introduction of some different scenarios, and in last part, an optimization algorithm that addresses the specified scenario(s) using linear programming is proposed. Finally, different results are analyzed and compared and some future work directions are drawn.

Contents

List of Figures

1 Introduction

There are many different definitions for cloud computing. Briefly, it can be defined: "Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs"[1]

Of course, this definition can change during time and new features can be added to it. Cloud computing is a new paradigm and will be complete in the future. There is a strong belief that one day cloud computing will be 5th utility (after water, electricity, gas, and telephony). [2] The computational resources (e.g. networks, servers, storage services) are provided to the customer on-demand by large data centers, which provide these resources on a pay-as-you-go basis. The main idea behind cloud computing is that it is not necessary that the computations being done in a physical machine, it also can be computed in cloud. For example, when a user needs more storage than physical machine, cloud can provide more storage.

It should be mentioned that cloud providers are not limited to only provide the infrastructure, they can also provide platform for developing new services. [3] Users should pay for using these services. Infrastructure as a Service (IaaS) is a cloud computing service model where consumers have access to virtual computers, storage, network infrastructure components, and other computing resources on demand and over the network. When an IaaS provider receives a request from a consumer, the provider has to allocate the request. Such a cloud service may consist of several datacenters located in different geographical zones and interconnected by dedicated network links. Deciding on the optimal resource allocation will therefore imply deciding on which datacenter to place each of the requested resources. The resource allocation problem is therefore an optimization problem where a solution is to be found that satisfies a set of constraints (e.g. QoS requirements) while one or more variables (e.g. deployment cost) are minimized. Different solutions for optimization problems of this kind have already been proposed in the literature. They differ in their application context (e.g. one datacenter or multiple ones), their goal (i.e. the variable(s) they try to minimize), considered constraints, and the like. Most of them deal with just VMs and do not take the network into account.

2 Outline

The organization of the report is as follows. First, state of the art in cloud computing and resource allocation is described. In the next part, different scenarios are discussed and summarized details about each scenarios are mentioned. Chapter 5 is dedicated to formulating the problem and presenting how it solves the model. Finally, it talks about comments and conclusions of the project in last chapter.

3 State of the Art

This chapter briefly presents the work that has been done so far in the field of cloud computing and specially resource allocation challenges. For better organization, this chapter starts with cloud computing background and continues with definition, different cloud classifications and some technologies behind the scenes of cloud computing. Finally, resource allocation concepts, which are the main topics of this project, along with a brief description of different subjects in resource allocation challenge are given.

3.1 Cloud Computing

3.1.1 Background

Cloud computing is a paradigm that is shaped during time by merging different varieties of computing fields.Although a lot of old and new concepts participate in the term "cloud computing", such as grid computing, utility computing and virtualization, but it has received a lot of attention in few past years. [3] The previous paradigms, such as client-server, cannot be practical anymore when the number of interconnected machines is increased, though they are used internally to shape new paradigms. Instead of using super-computers for doing complex tasks, there is an opportunity to rely on virtual super-computers. This can provide a large amount of resources for requirements. It is called as cluster computing. [4]. In cluster computing a number of servers are connected together and work as one integrated resources, and users see it as one entity.

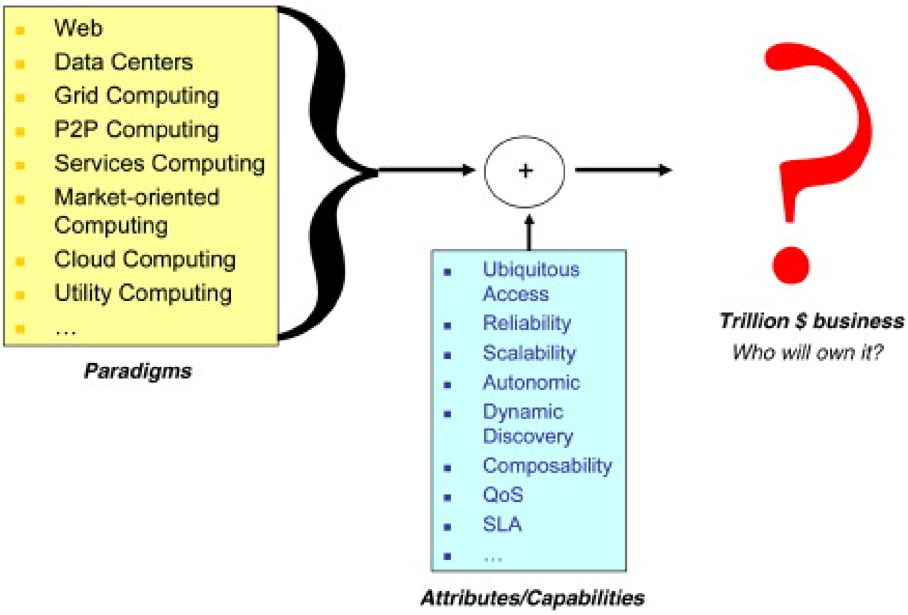

Figure 1: Different paradigms [2]

The other paradigm which is used for decades is grid computing. In grid computing a large number of machines are connected together as a single entity. Many grid and cluster commuting features are the same, but allocating resource infrastructures in different geographical places make it unique. It should be mentioned that two concepts of cloud and grid computing are used interchangeably sometimes but, they have some differences. Although it is hard to distinguish between them, and it should be mentioned a lot of other paradigms indirectly influence cloud computing. In addition to new paradigms that are presented, new technologies also are appeared. One of these technologies is virtualization. Virtualization refers to create a virtual (rather than actual) version of something. It can be an operating system, a server, a storage device or network resources [4]

3.1.2 Virtualization

Everyone is familiar with the concept of partitioning (by dividing a hard disk into different partitions). Similar to that, OS Virtualization gives the opportunity to divide storage, and run different OS (operation system) images at same time. This was just one example of virtualization.

Using virtualization helps to reduce hardware and saves cost and energy in computer networks. This is one of the best advantages of virtualization. Moreover, virtualization separates service requests from the underlying physical delivery of service.

Virtualization has existed in the IT field at least from 1970[10] but introducing hardware virtualization of x86 in 1998 was an enormous evolution in cloud computing. In the beginning, virtualization was just about memory, but then it deployed in other infrastructure layers such as application and operation system. Figure 4 compares two architectures, one with virtualization and one without it.

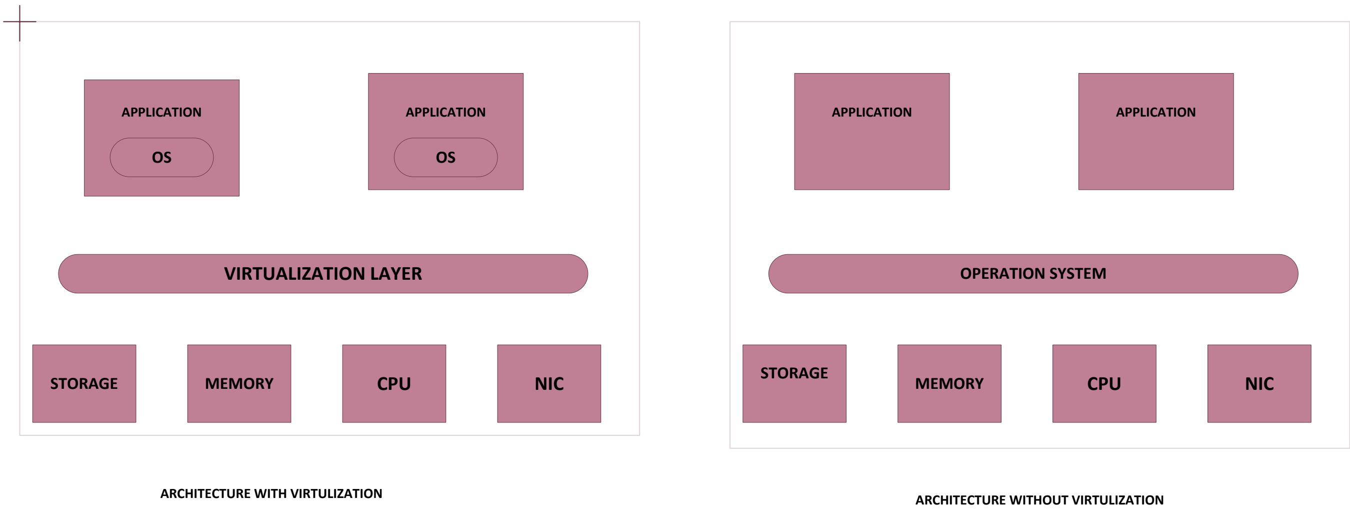

Figure 2: The architecture of virtualization

As seen from Figure 2, the virtualization layer is placed between hardware and operating system. In the right architecture (without virtualization) the operating system depends on hardware and that is why it uses specific device drivers for adaption, but in the architecture with virtualization, all of storages, memories and other resources are considered as a big pool, on top of which are virtual machines. A virtual machine is a piece of software installed on hardware making the hardware abstract for operating system.

Key features of virtual machine are categorized here:
Partitioning
Running multiple virtual machine simultaneously on a single sever
Isolation
Each virtual machine is independent from others in one server
Encapsulation
Virtual machine encapsulate all of system (hardware, OS and apps) in files
Hardware Independence
Running virtual machine in any server without changing hardware configuration

Server utilization in old architecture (without virtualization) is between 5-15 percent while in new architecture is between 60-80 percent. There are two different type of virtualization in practice: full virtualization and paravirtualization. The difference is about the quest OS can be run on VM with or without any modification. In full distribution, it can be run without any change but, in the other it should be modified before to run.

In addition to these two types of virtualization, there are two other models: bare-metal, hosted. Before talking about these types it is necessary to look at the virtual layer in more details. There are two main components in this layer: hypervisor and virtual machine monitor. VMM provides a virtual machine to the client's OS. The role of hypervisor is to virtually partition resources. In bare-metal, hypervisors run directly on the physical resources and the virtual machine manager provides a connection between client operating system and hypervisors, but in the hosted virtualization, as it is apparent from its name, a host operating system runs on top of the physical hardware. In this virtualization there is no hypervisor, but there are virtual machine managers that provide links between VMs and OS. [10]

3.1.3 Definition

Though cloud computing has received considerable attention during recent years and a lot of different definitions have proposed, but these definitions focus on some aspects and forgot to address other features of this concept. In [1] the author tried to define cloud computing as: "A large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing also for optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs." By looking at different definitions, it is clear that there are some common characteristics that all can agree with that: on-demand self-service, broad Network Access, Resource Pooling, Rapid Elasticity, and Measured Service. [1] [5]

On-demand self-service. This definition consists of two parts. First, the computing service is provided to the user on demand. Second, users can get services automatically without requiring human interaction.[5]

Broad network access. It simply means that the user can access cloud services on different varieties of platforms. (Mobile phone, laptops,...) [5]

Resource pooling. A cloud provides all resource requirements such as memory, storage in a large pool. According the multi-tenant model, for example, more than one user can use one server. It is not necessary to put all resources in the same data centers they can be in independent locations. Virtualization and dynamic resource allocation are necessary requirements of this concept.[5]

Rapid elasticity. Simply, this concept addresses that computing resources appears to be infinite to the users. [5]

Measured Service. User of cloud services should pay according to some unit such as usage of services per months. This concept is regarded as utility computing concept. cite5

3.1.4 Cloud Classification

Cloud can be classified as two big models:

Deployment model

Private cloud. The cloud infrastructure is operated only for an organization or company. It can be managed by the user or a third provider.[5]

Public cloud. The cloud is available to the general public. By paying, users can access to the cloud services. [5] It should be mentioned that public cloud should satisfy some level of quality of service like latency or delay.

Hybrid cloud. The cloud is a composition of two or more clouds (private, or public) .[1][5]

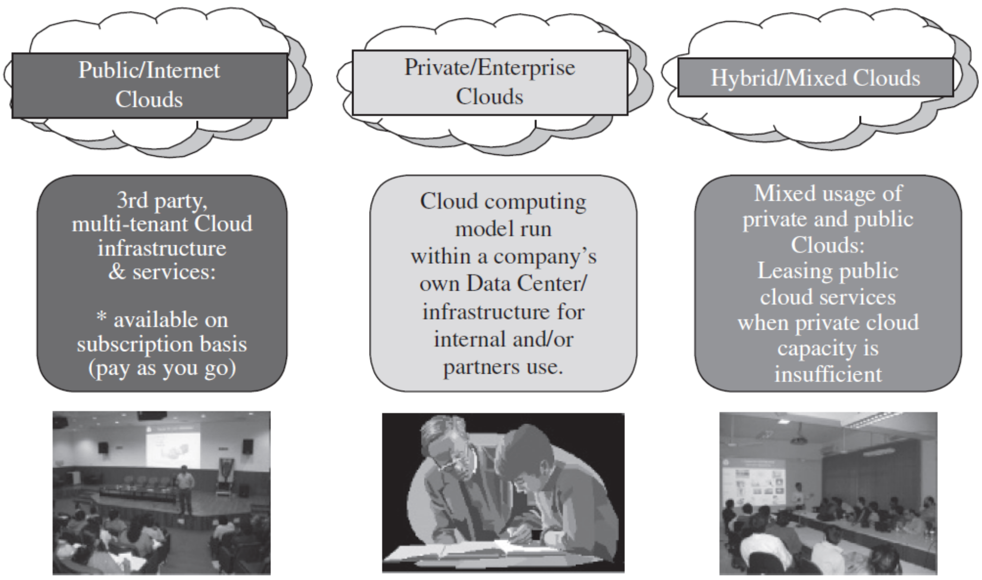

Figure 3: Private, Public and Hybrid clouds [21]

Server Models Cloud Software as a Service **(SaaS)**. A cloud can give services to users by providing software running in cloud. For example Google Docs is an application running by providers in its infrastructure and users can access it by the Internet. But actually this idea (SaaS) is not new and existed before [5]

Cloud Platform as a Service **(PaaS)**. The famous example is Windows Azure. Instead of providing software as service in this case, cloud provides platform for customers. By using this service users can create their own software though still they cannot have access to hardware.[5]

Cloud Infrastructure as a Service **(IaaS)**. The last service that cloud provides is infrastructure as service. Cloud has some data centers that have some virtualized servers. These data centers can satisfy a couple of computing requirements of users. Clients can use these infrastructures by paying. These infrastructures consist of storage, processor and network . Amazon EC2 is a famous example of IaaS.

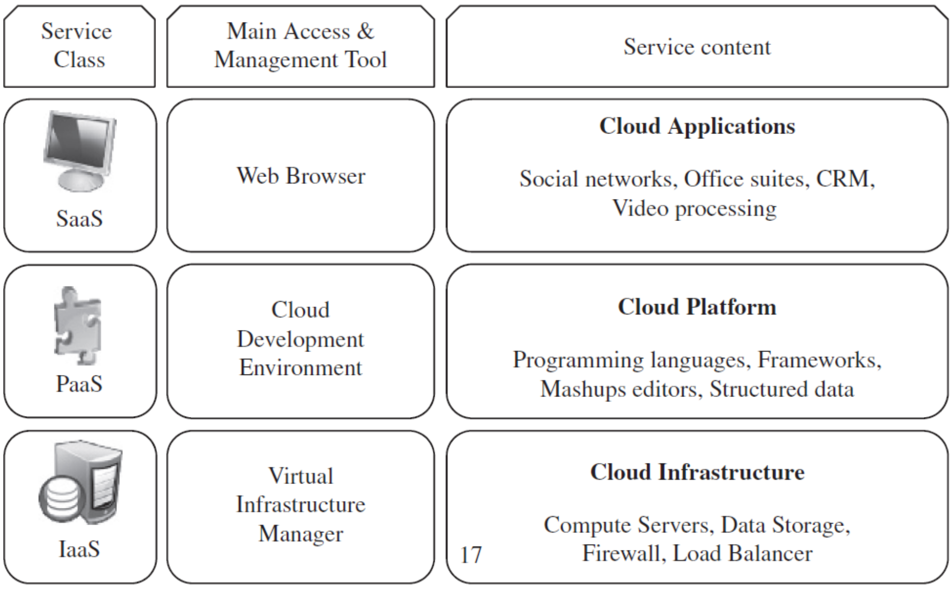

Figure 4: Service models [21]

3.2 Resource allocation

This section organized as follows: first, a general definition of resource allocation is given. Then, more details about the current challenges are given and finally some available solutions are presented.

3.2.1 Resource allocation

Resource management and allocation is one of the most important issues in cloud computing. The cloud operator should provide applications with automatic means for acquiring available resources, aiming at the best utilization of the resources. A Resource Allocation System (RAS) is a mechanism in cloud computing that ensures the application's requirements are correctly handled by the provider's infrastructure. In addition to this guarantee, a RAS also monitors the available resources for minimizing the operational costs.

The way resources are modeled and presented to the user is essential to the cloud's operation. The modeling can be done in various abstraction levels. Cloud resources can be considered as any resource that the developer may request from the cloud. Examples of cloud resources are network requirements such as bandwidth, or computational requirements such as CPU and memory. Resources in the cloud are usually hosted in a data center and are shared among many clients. Therefore, a RAS should be able to dynamically allocate these resources to developers. Resource modeling, cloud resources, and developer requirements can be seen as the input to a RAS. These inputs are discussed next when we present the research challenges in resource allocation.[6],[18][19]

3.2.2 Research challenges in resource allocation

The challenges in a RAS can be divided into four groups: resource modeling and description, resource offering and treatment, resource discovery and monitoring, and resource selection. The first two challenges are related to the conception phase, in which the cloud provider should model the resources and decide about how they will be offered. The last two challenges are faced in the operational phase, in which the resource provider should provide a discovery service to attend requests and to select available resources to serve the requests. [6],[18][19]

3.2.3 Resource modeling and description

Resource modeling and description specifies how the cloud defines its infrastructure resources. Management, control and optimization of resources depend on the modeling. Therefore, the modeling is an essential feature of a cloud. There are several existing standards such as Resource Description Framework (RDF) and Network Description Language (NDL) which provide resource definitions for network and computing resources. However, in the Cloud, virtual resources should also be defined in a similar way existing resources are defined. Moreover, resources can be defined in different details level. The level of details affects the optimization in the cloud. For example, if a model is very detailed, the optimization may become very difficult, as many parameters should be taken into account.

There are many solutions to resource modeling and description. Resource modeling is closely related to interoperability which is a bigger challenge in cloud computing. The goal of interoperability is to facilitate the integration of different cloud applications. To this end, standardization and open APIs are very attractive for interoperability.

An approach for standardization of the definition of resources is Elastic Modeling Languages, which is a set of three languages based on Web Ontology Language. These three languages are Elastic Deployment Modeling Language (EDML), Elastic Computing Modeling Language (ECML), and Elastic Management Modeling Language (EMML). EDML is used to describe common servers in datacenters such as web and application servers. ECML provides a structural description for applications. Finally, EMML allows information retrieval on ECML and EDML elements.

A different language for modeling resources is Application Descriptor Language (ADL). Resources in ADL can be physical or virtual. In ADL resources are described using three kinds of descriptors. The component descriptor describing network applications such as a database server. Component descriptors can have optional attributes. For example, the migratable attribute specifies that a component can be migrated to another host. The assembly descriptor describes how components can be assembled together. Assembly descriptors in a sense specify the structure of a component. The package descriptor provides a directory for sharing reusable components among different applications.

The last group of languages used for describing resources are XML-based specification languages. Examples of these languages are Service Modeling Language (SML) and Service Modeling Language Interchange Format (SML-IF). SML is a set of interrelated XML documents describing different aspects of a resource and a set of constraints establishing their correctness. The constraints are applied using XML schemas or a set of rules implemented in XPath. SML-IF enables interoperability between and exchange of SML documents. [6],[18][19]

3.2.4 Resource offering and treatment

The next step after resource modeling is how to offer the resources to the developers. Resources are offered through interfaces allowing the user to specify their application's requirements. Under the hood, a RAS should handle resources at lower levels. To handle resources, the cloud provider implements solutions to control all the available resources in the cloud. They may also employ virtualization solutions offered by vendors such as VMWare.

As mentioned before, in addition to traditional network requirements, cloud providers should consider other requirements specific to the cloud. Examples of these requirements are topology of the network, jurisdication, and node proximity. The topology of network requirement provides the developers with the option to configure network nodes and their relationships to each other. Jurisdication is related to where data have to be physically kept. For example, due to copyright rules, some files should not be located at specific places. Finally, node proximity specifies the maximum and minimum distance or delay between nodes. [6],[18][19]

3.2.5 Resource discovery and monitoring

Resource discovery is the process in which the cloud resources are found. Moreover, the cloud should be able to answer questions about node proximity and network traffic. Given the dynamic nature of resources in the cloud, resource monitoring should be a continuous process by which the state of resources is perpetually checked. It should, however, be mentioned that care should be taken not to waste too much bandwidth on refreshing the status of resources.

Resource monitoring can be passive or active. In the passive mode, there is a cloud entity which is active and continuously collects information about resources. Resources are polled by messages sent from entity. The entity can also directly read the needed information from a resource. In contrast, in the passive monitoring type, resources act as agents and send information to a central cloud entity. In a cloud, both active and passive alternatives may be employed at the same time.[6],[18]

3.2.6 Resource selection and optimization

Usually when a resource is requested, there are multiple options available. Therefore, the cloud should select the resources which satisfy all the user requirements and minimize the cost of cloud infrastructure. In virtual networks, for example, the optimization is how to select the best mapping from virtual resources and the underlying substrate network.

It should be mentioned that due to the dynamic nature of the cloud, resource selection and optimization is not trivial. The resource selection can be implemented using an optimization algorithm such as the traditional linear programming or some newer variants such as Lyapunov Optimization. Some AI techniques such as ant colony and game theory are also viable options for optimization in the cloud.

The resource selection in the cloud can be done a priori or a posteriori. In a priori mode, as its name implies, the optimization is done before selecting resources. Therefore, the selection is already optimal. For this, the selection procedure should consider all the variables and requirements and decide accordingly. In a posteriori mode, first, an initial selection, which is suboptimal, is made first and then the resources are continuously optimized. For example, more memory or storage can be added to a VM along the way.[6],[18][19]

3.2.7 Solutions to resource allocation

The resource allocation problem is not new, and there are currently various solutions to the challenges reviewed in the previous section. Many of these solutions focus on the optimization of energy consumption, time-response management, and resource allocation in virtual networks.

Nowadays, energy management is a global design criterion, especially in cloud computing. Data centers and applications are growing very quickly. This makes the lowering of energy consumption very challenging. Energy saving can be achieved through many strategies such as moving the workload of VMs or turning off idle machines.

An energy saving solution is Datacenter Control Algorithm (DCA) which uses queuing information to implement energy management in virtualized environments. The queuing information is used to make online decisions based on the workload. When a server's workload is low, DCA turns the server into the inactive mode to save energy. DCA fulfills client requests using an admission module which verifies if a new request can be answered. Admitted requests then are routed between the available servers to select active servers with the smallest request queues. There are also a number of similar solutions which all try to achieve a compromise between energy consumption and performance by switching off idle servers. There solutions use monitors to control applications performance and energy consumption.

Response time management, which is an important concept in Cloud Computing, is related to how a RAS allocates resources to achieve a good performance. In grid computing, Makespan is the amount of time from the initiation of the first task to the time the last task finishes. Due to the dynamic nature of grid resources, achieving the makespan property is challenging. In Cloud computing, we also face the same problem. There are a number of scheduling algorithms, such as Min-min, Max-min, and XSufferage, whichi are applied when some tasks take more time than their intended time.

The last solutions related to resource allocation in the cloud are related to network virtualization, which is a new research topic in network. The main problem in network virtualization is how to allocate virtual resources over a substrate network. A way to allocate virtual resources is to use graph resource modeling. In this approach, the cloud infrastructure is seen as a graph. Virtual network requests are also modeled as graphs. Requirements such as memory and CPU requirements can then be seen as values associated with edges and nodes of this graph. With this model of resources and requests, resource allocation can be seen as a simple mapping from the request graph to the substrate graph.

Given the graph model, an allocation algorithm that maps a virtual network to its substrate network should consider the constraints such as CPU and memory requirements, and try to optimize an objective function based on these constraints. For example, an approach is to reduces the allocation problem to use mixed integer programming to solve the problem in polynomial time. There are still several challenges in resource allocation in virtual networks. For example, dynamic allocation algorithms are being developed which are based on the bandwidth consumption of virtual networks. These algorithms can be implemented using game theory or adaptive feedback control theory. Furthermore, the use of heuristics to solve these NP-hard problems is essential.[6],[18][19]

4 Problem Description

A description of exactly what the problem is, how to model the problem in different scenarios, and some implementation notes

4.1 Introductions

Infrastructure as a Service (IaaS) is a cloud computing service model where consumers have access to virtual computers, storage, network infrastructure components, and other computing resources on demand and over the network. In this particular case, Consumers request virtual machines (VMs). Virtual machines started to replace the original hosts in cloud. Using virtual machines allows the clients to meet their computational requirements. A virtual machine (VM) is a "simulation of a machine (abstract or real) that is usually different from the target machine (where it is being simulated on). Virtual machines may be based on specifications of a hypothetical computer or emulate the architecture and functioning of a real world computer.[20]"

When an IaaS provider receives a request from a consumer, it has to decide the optimal allocation . Such cloud may consist of several datacenters located in different geographical zones and interconnected by dedicated network links. Deciding on the optimal resource allocation will therefore imply deciding on which datacenter to place each of the requested resources.

Obviously the resources in a cloud are limited and always there are some constraints like latency, cost, and energ. For example, the requested resources will all be used by one service or application and will therefore be interrelated (e.g. two VMs may need to be deployed close to each other in order to minimize the latency when communication with each other). The resource allocation problem is therefore an optimization problem where a solution is to be found that satisfies a set of constraints while one or more variables (e.g. deployment cost) are minimized. In the following part the exact model of problem with all its details is explained.

4.2 Model description, assumption and simplification

In this section, a general model of the system is described. As seen in Figure 5, a cloud consists of a number of nodes. The term node represents a geographical zone (India, Europe, Asia) that consists of physical data centers. According to the author defined in [9]"A data center (sometimes called a server farm) is a centralized repository for the storage, management, and dissemination of data and information. Typically, a data center is a facility used to house computer systems and associated components, such as telecommunications and storage systems. Often times, there are redundant or backup power supplies, redundant data communications connections, environmental controls, and security devices". The nodes are marked by numbers (see Figure 5).

In addition, a node can just be a router without any data center and just connect other nodes to each other. The nodes are connected to each other depending on the network architecture. In Figure 5, the lines indicate the connections between nodes. It assumes that the system is not fully connected (in our case study). According to the definition, a computer cluster is a

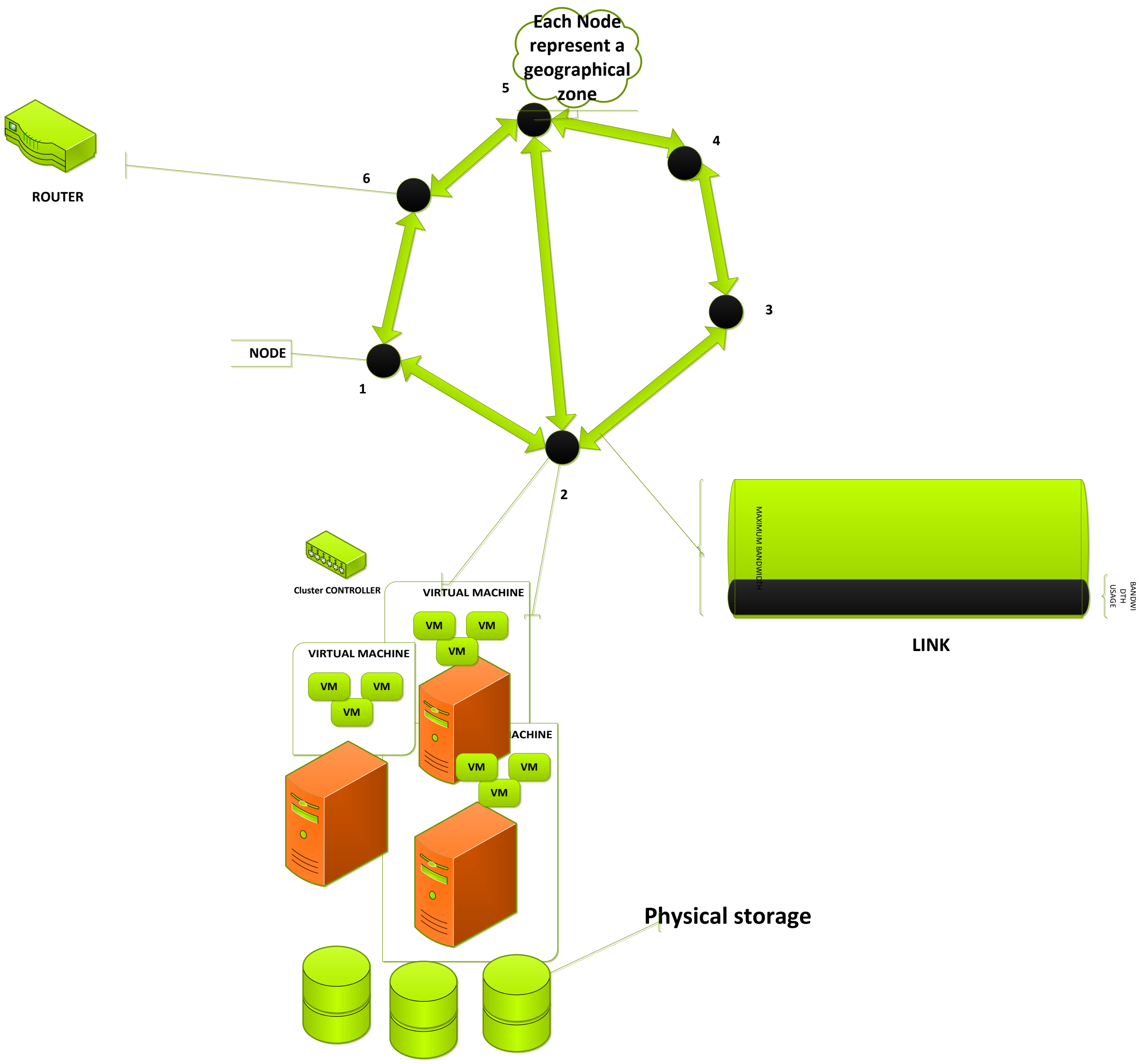

Figure 5: Model description

group of physical computers (host) that work together, and they can be considered as one entity. Each host has its own specifications. For example, a machine with a 2GHz CPU with 2GB of RAM. When the user requests a virtual machine, for example, to run a program on Linux, it needs 2 GB of RAM and a CPU more than 2GHz and 20 GB for storage. So it can be easily calculated that how many VM can be allocated in a datacenter. (See Figure 6)

If the nodes are heterogeneous, then each node has own parameters: CPU settings (in GHz) and the execution cost. It is important to mention that the term of cost in cloud data center consist of a wide range of parameters. As author mentioned in [14], the cost in data center can be categorized as in the table below.

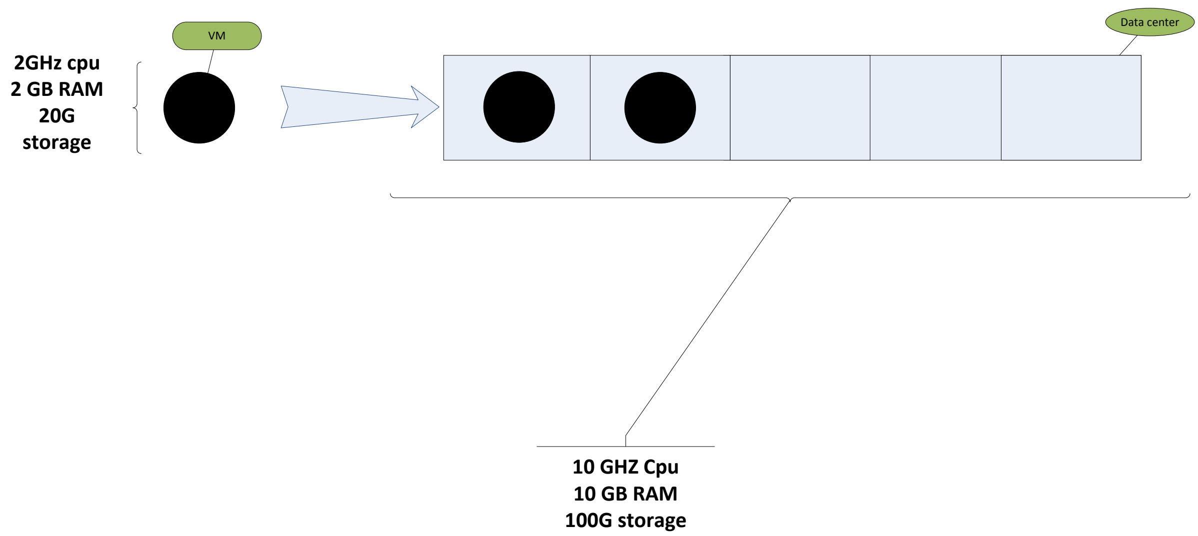

Figure 6: Data Center

Amortized Cost	Component	Sub-Components
~45%	Servers	CPU, memory, storage systems
~25%	Infrastructure	Power distribution and cooling
~15%	Power draw	Electrical utility costs
~15%	Network	Links, transit, equipment

Table1. The cost in data center [14]

In this particular issue we limited cost to links and CPU. Nodes are connected by links. The term link in communication network refers to a channel that connects at least two devices. For example, fiber optic can connect some communication devices together. A link can be physical or logical. Each link has some features. One of these features is bandwidth which is the rate of transfer bit (the amount of data is passed from one point to another point in a period of time). Heavy tasks such as vidwo streaming demands more bandwidth than speech or data. When a data passed through a link, it will allocate some bandwidth (see Figure5) and as every link has limited bandwidth it can carry certain amount of data in period of time. In Figure 15 the main view of system is presented. Users request services (VMs) from a cloud broker. The responsibility of broker is to deploy the client's requests in cloud. The broker tries to find an optimal deployment that optimizes costs.

After explaining the system, now it is time to present some scenarios of how we should deal with this system in order to minimize the cost of system. Two different approaches are offered here, but the main focus is on the second one that we went to details and tried to implement system. The difference between the two scenarios is how requests will be defined. In the first scenario, we are faced with interrelated requests, but in second one the requests are independent.

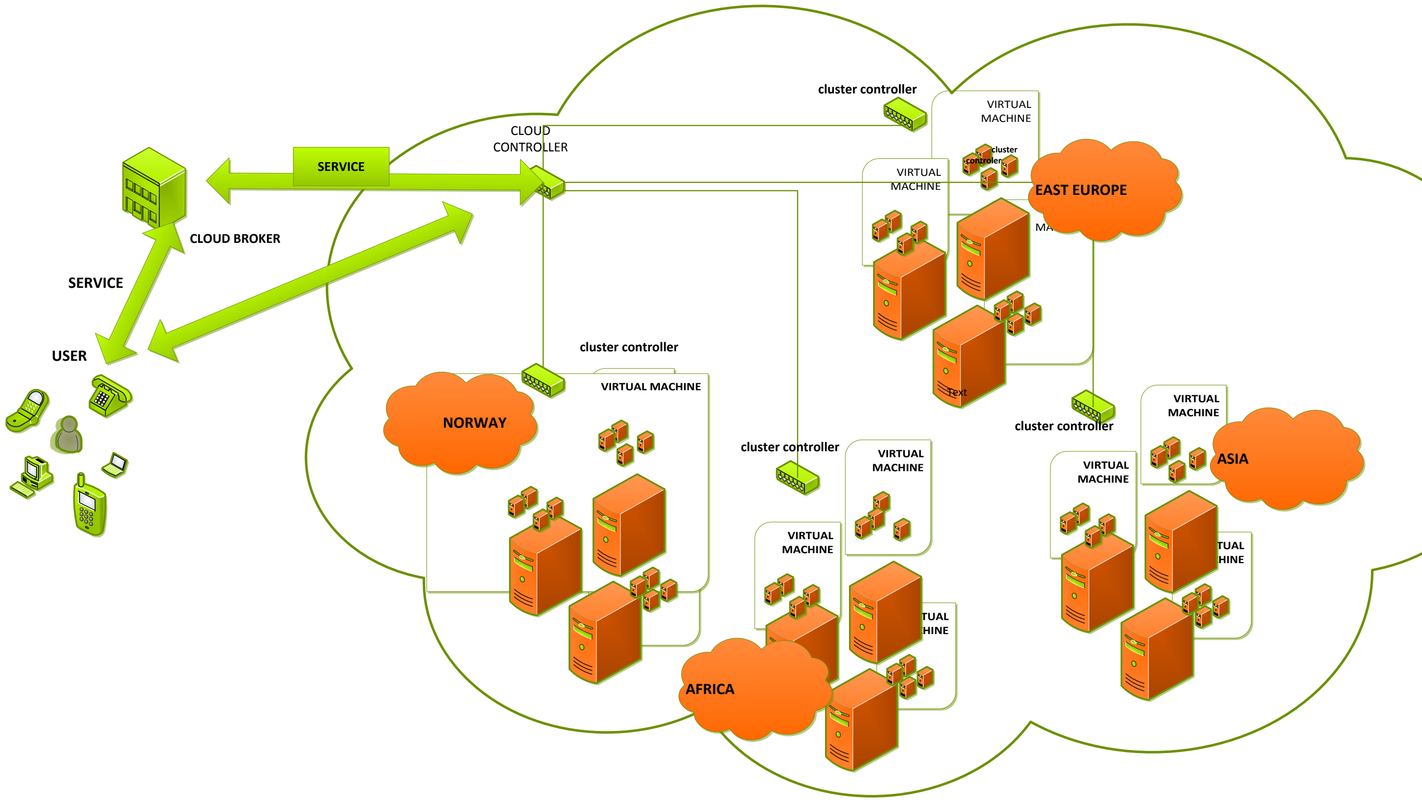

Figure 7: General model [15]

5 Related work analysis

5.1 Scenario 1 Interrelated VMs

As seen in Figure 8, in the first scenario, the VM requests are interrelated to each others. For example a user needs to run a large program that needs considerable resources, such as storage, different operation systems, and memory. These components should be connected in a specific map. The problem here is how to allocate interrelated VMs in physical nodes. In [13] Mate faced with the same problem about software deployment in networks. We use the same procedure as his approach to solve this problem. this problem

It is assumed that $\mathbf{V}$, the number of virtual machine, should be allocated in $\mathbf{N}$ nodes. The virtual machines are interrelated. The VMs can be connected by $\mathbf{K}$ collaborations. $\mathbf{N}$ nodes are physically connected. Each VM and collaborations has own cost (as explained in previous section). It is assumed some VMs are restricted to some nodes. For example, some data or some special programs just can be accessed in special nodes. One of the reasons for restriction can be security. For example, some secure data have to be kept in Norway only. Mate called these components (VM) bounded components.

For two VMs allocated in the same node, there is no commutation cost, and the communication cost is just considered for VMs placed in different nodes. In Figure 8, VMs 1 and 2 are allocated in node 1 and VM3 is allocated in node 2, and finally VM4 in node 3. Therefore, the communication cost between 1 and 2 will be zero because they are in same node (1)

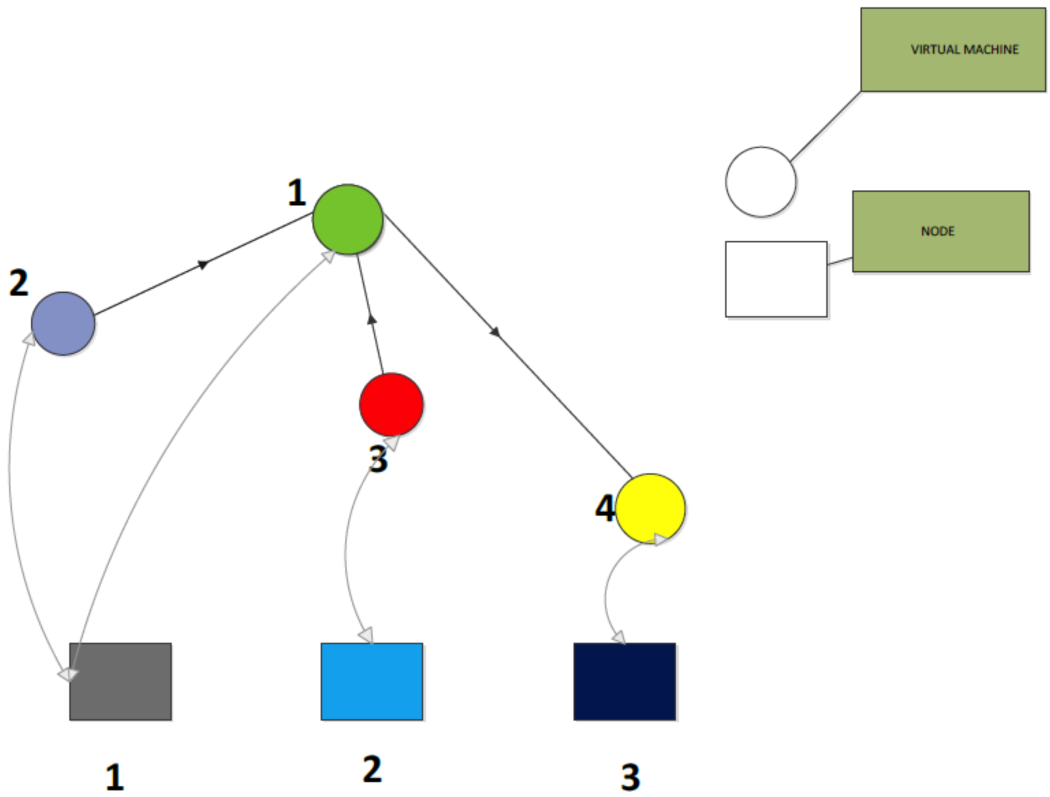

Figure 8: Interrelated Virtual machine model

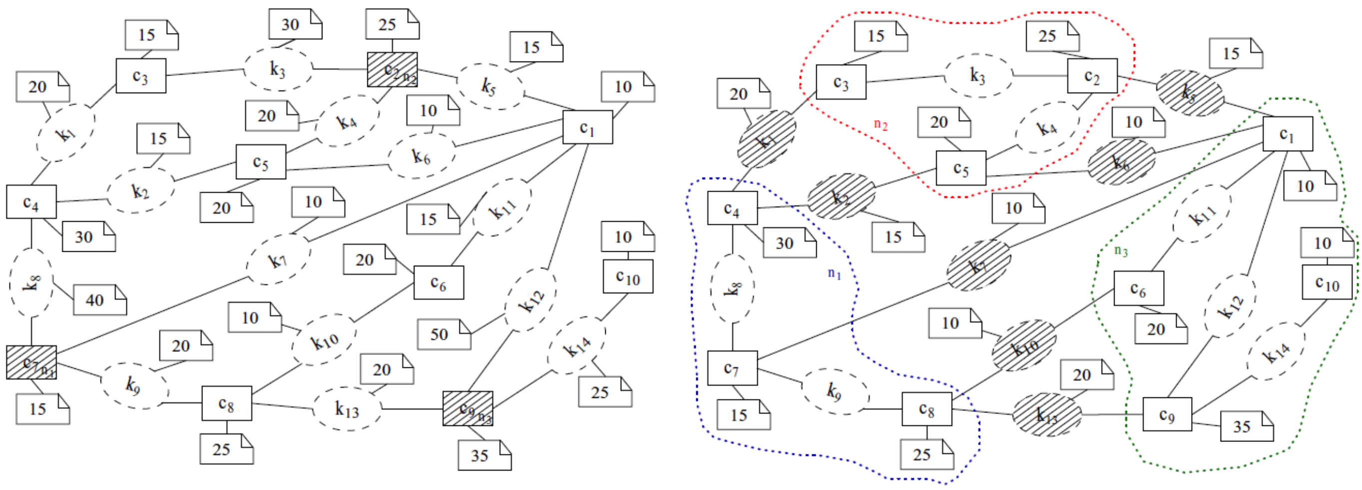

Figure 9: Allocated components to nodes in interrelated model [13]

and the cost here is just consist of cost of each VM, but for the commutation between VM1 and VM3 there are some cost involved because they are in two separate physical nodes. For solving this problem (deploying service components in network) Mate applied heuristics and a nature-inspired optimization method, called the Cross Entropy Ant System (CEAS) In order to make the problem clear some figures are added.Figure 9 shows that how these components allocated in 3 nodes. As can be seen, some components (the black squares) are bounded to special nodes. For more details about the algorithm see [13].

5.2 Scenario 2 independent VMs

In the second scenario, it is assumed that each request consists of one VM that has its own features such as bandwidth and CPU usage. The requests are independent of each other. Before going to details, it should be mentioned that in [16] Otto faced a similar problem. In peer-to-peer networks each computer acts like a client or server and there is no need for a central server. Locating desired services in such a system is a big challenge. For example, due to lack of network bandwidth access, achieving a high quality video conferencing is impossible in some cases. Otto considered a model that consists of a number of clients and servers and links between them. He defined three resources: client, transport and peripheral. A client is the first resource in a path. Transport resources provide transport service. They act like a router or switch and finally peripheral resource is the last resource in the path and the user normally searches to reach this resource. Otto has assumed that there are at least two resources in path, one client and one peripheral. The other concept in this system is profile. There are two classes of profiles: user profile that contains QoS parameters specifying constraints and preferences in the request, i.e. QoS objectives.

Service profile contains QoS parameters specifying limitations and capabilities of a resource. Qos requirements are delay, bandwidth and type of resources. To make the system simpler, the author assumed that between each resource only on path exists. He proposes a swarm-based distributed multi-criteria optimization algorithm that can find optimal paths of resources in a large and complex network. In summary, in this problem they faced a peer-to-peer network and had to find the best path between a client source and the peripheral.

6 Offered solution

In this scenario, it has been assumed there are a couple of requests for a VM that are independent of each other. A cloud broker acts between clients and cloud. It receives requests and will allocate them using specific data centers. Therefore, in this scenario, there are many requests that should be allocated. Each request consists of one VM and they are not interrelated. There are some differences between this work and Otto. He considered peer to peer networks and he used ant colony to find the best path between two nodes, but in this case a cloud system with different data centers is considered. Therefore, the problem is the same but, the case study is different (peer to peer network vs. cloud) and also the approach that is used completely different. Linear programming (LP) which can be applied to validate results of Otto's work is used in this scenario. The results by optimization are not precise, but the results by LP are completely accurate, so this work can be seen as the completion of Otto's work.

6.0.1 A simple network

The aim of this part is optimization of a very simple network to verify that this solution gives us an optimal answer. It is assumed that there is a network with three nodes and each node has own specifications and there are some links between nodes as seen in figure 10.

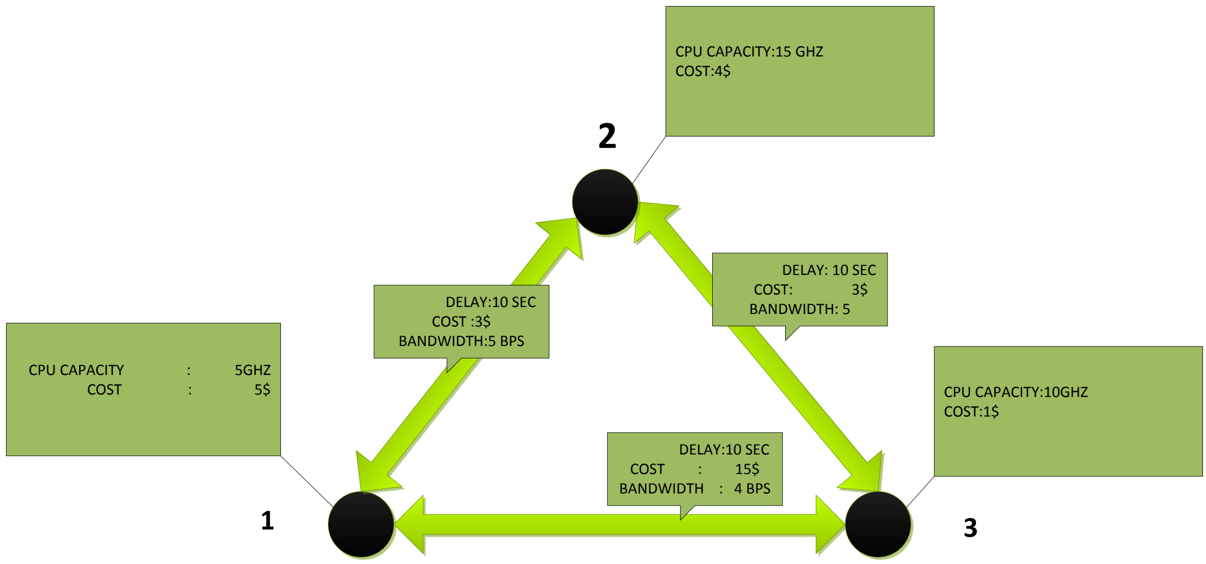

Figure 10: A simple network model

To make it simple, it is assumed that there is just one client that requests one VM. The goal is allocating VM in a node in order to have optimal results. This request has these specifications: Destination: =1 (The location of user) ,BwR := 3, CapacityR := 5 DelayR := 5. (The user is located in node 1 and the question is: which node is the best node to put a VM on in order to minimize the costs. There are various solutions for this problem. All possible solution are shown below:

In the table below the cost of each solution is presented:

	Solution1	Solution2	Solution3	Solution4	Solution5
Cost	5*4+3*3=29	5*4+3*3+3*15=74	5*1+3*15=50	5*1+3*3+3*3=23	5*5=25

Table2. Cost of different solutions

Solution 4 is the best and can minimize the cost and also satisfy the constraints. AMPL is used in this project (see Appendix 1). This small example showed that finding the best answer is not trivial and definitely for a big network and for a lot of requests it is more challenging. Therefore, it is required to use AMPL to solve this solution.

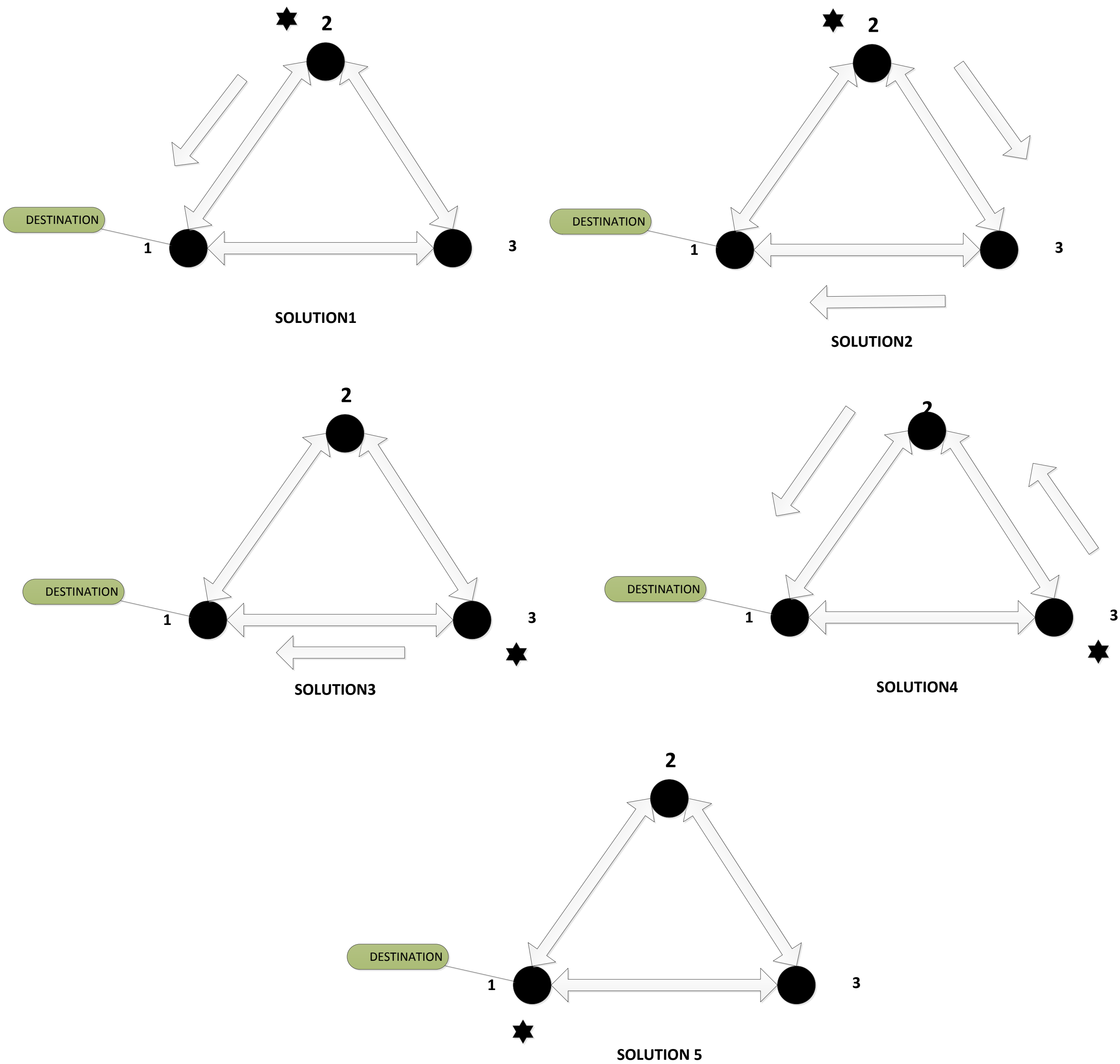

Figure 11: Different solution for a simple network

6.0.2 Formulation

The objective is minimizing cost:

MIN $(\sum_{i,j\in ARCS}\sum_{k\in REQ} \boldsymbol{COSTB(i,j)} * \boldsymbol{BWR\ (k)} * \boldsymbol{\alpha} + \sum_{m\in NODES}\sum_{g\in REQ} \boldsymbol{COSTC(m)} * \boldsymbol{CAPACITYR(g)} * \boldsymbol{\beta}\ )$

What is α **?**

If the Link (i ,j) is used in that request then α =1 otherwise its 0

What is $\boldsymbol{\beta}$**?**

If the Node i is used in that request then β =1 otherwise its 0

Then we should put some constraints to satisfy the problem specifications.

Each node has limited capacity so this constraint should be satisfied:

$\sum_{i\in REQ} \boldsymbol{CapacityR(i)} * \boldsymbol{\beta} \leq \boldsymbol{Capacity}$**(node)**

Each connection link also has limited bandwidth so:

$\sum_{i\in REQ} \boldsymbol{BWR(i)} * \boldsymbol{\alpha} \leq \boldsymbol{BW}$**(link)**

And each connection link has an acceptable delay:

$\sum_{i\in REQ} \boldsymbol{DelayR(i)} * \boldsymbol{\alpha} \leq \boldsymbol{Delay}$**(link)**

6.0.3 Telenor network

As this specialization project is done in Telenor. In this part, the main attempt is to simulate the Telenor network. In Figure 12, the Telenor network topology is shown.

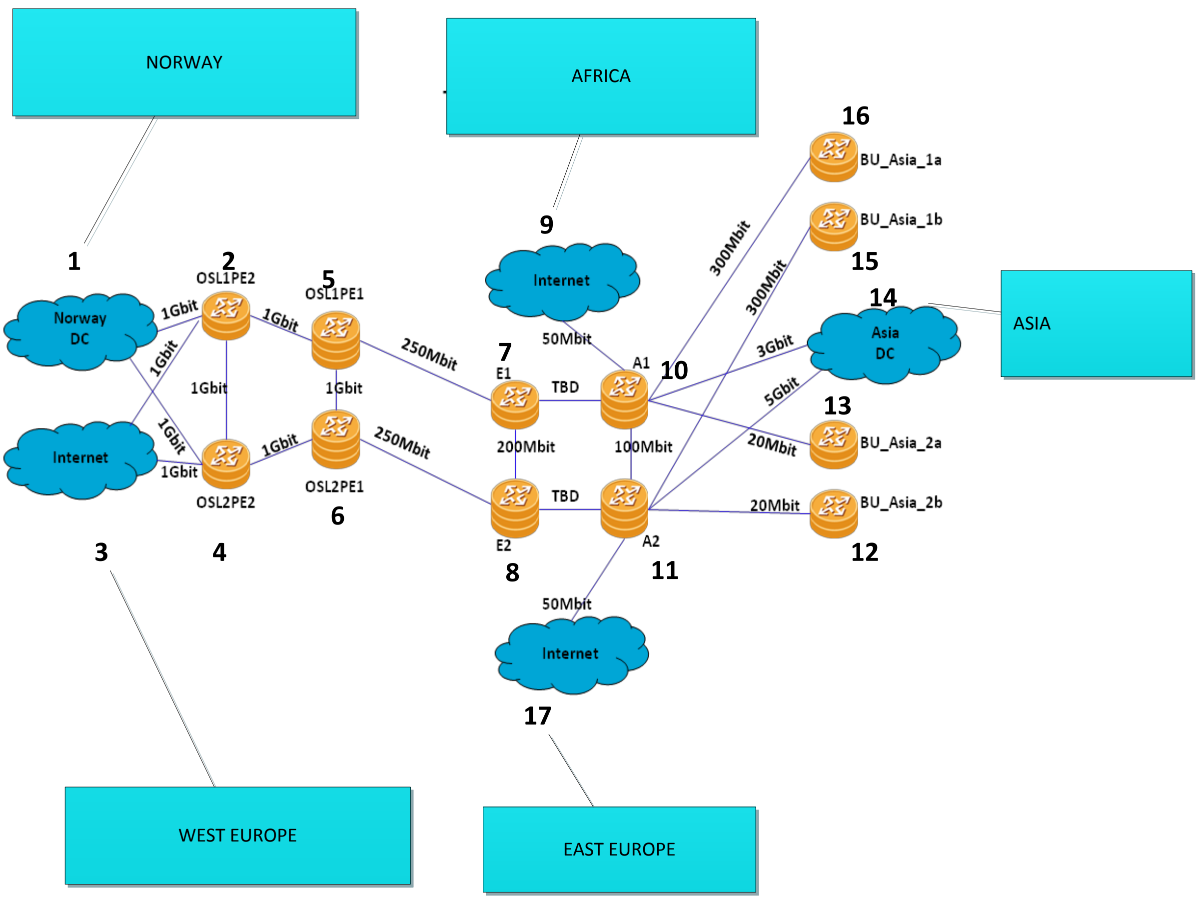

Figure 12: Telenor Network

As can be seen from Figure 12, the network consists of 17 nodes which are connected by communication links that have their own delay, bandwidth and cost. The Network consists of 5 big data centers. We already know the cost in Asia is less than other places, but for a user in Europe in addition to cost, other parameters such as latency, are also important. The results from 500 requests are given in Appendix 3.

7 Comparison and Conclusion

In Figure 13 and 14, the results of the same network (Telenor -17 nodes) for two different scenarios are given. In the first scenario, minimal cost for different requests without delay parameter is calculated and in the second one delay is considered as an additional parameter. Comparing two diagrams shows that when a new parameter is added, the cost is increased (See figure 15)

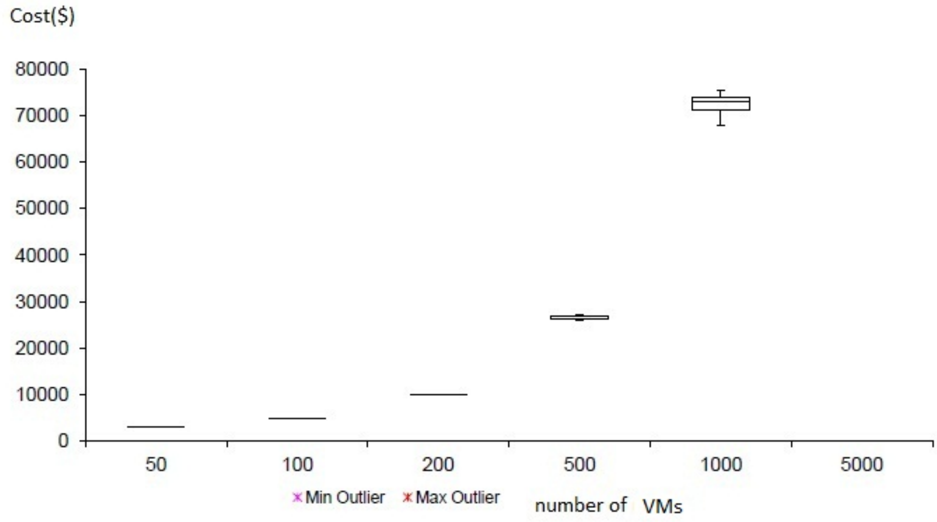

Figure 13: Cost without Delay

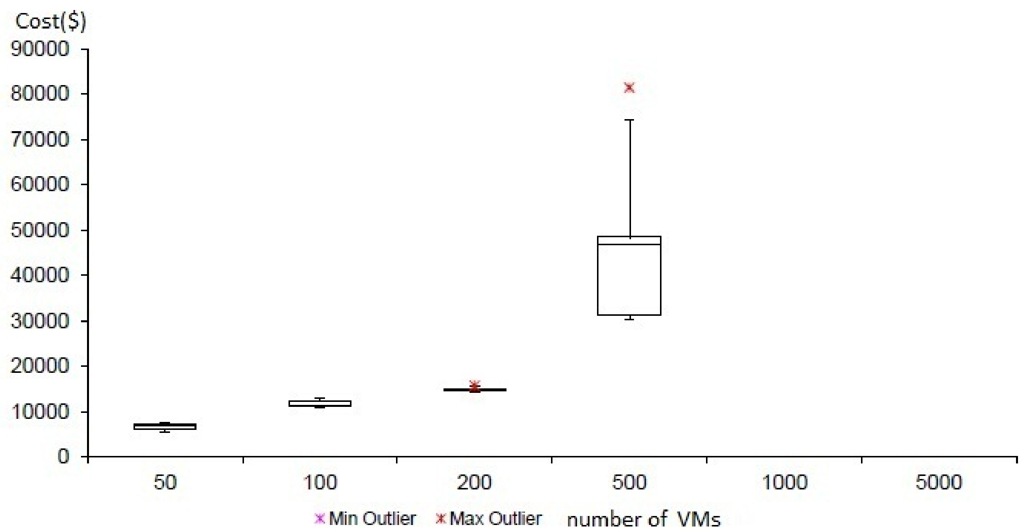

Figure 14: Cost with Delay

In addition to cost, the time of optimization for finding the best solution when it deals with more constraints will be increased. (See Figure16, 17)

Figure 18 the comparison of two diagrams.

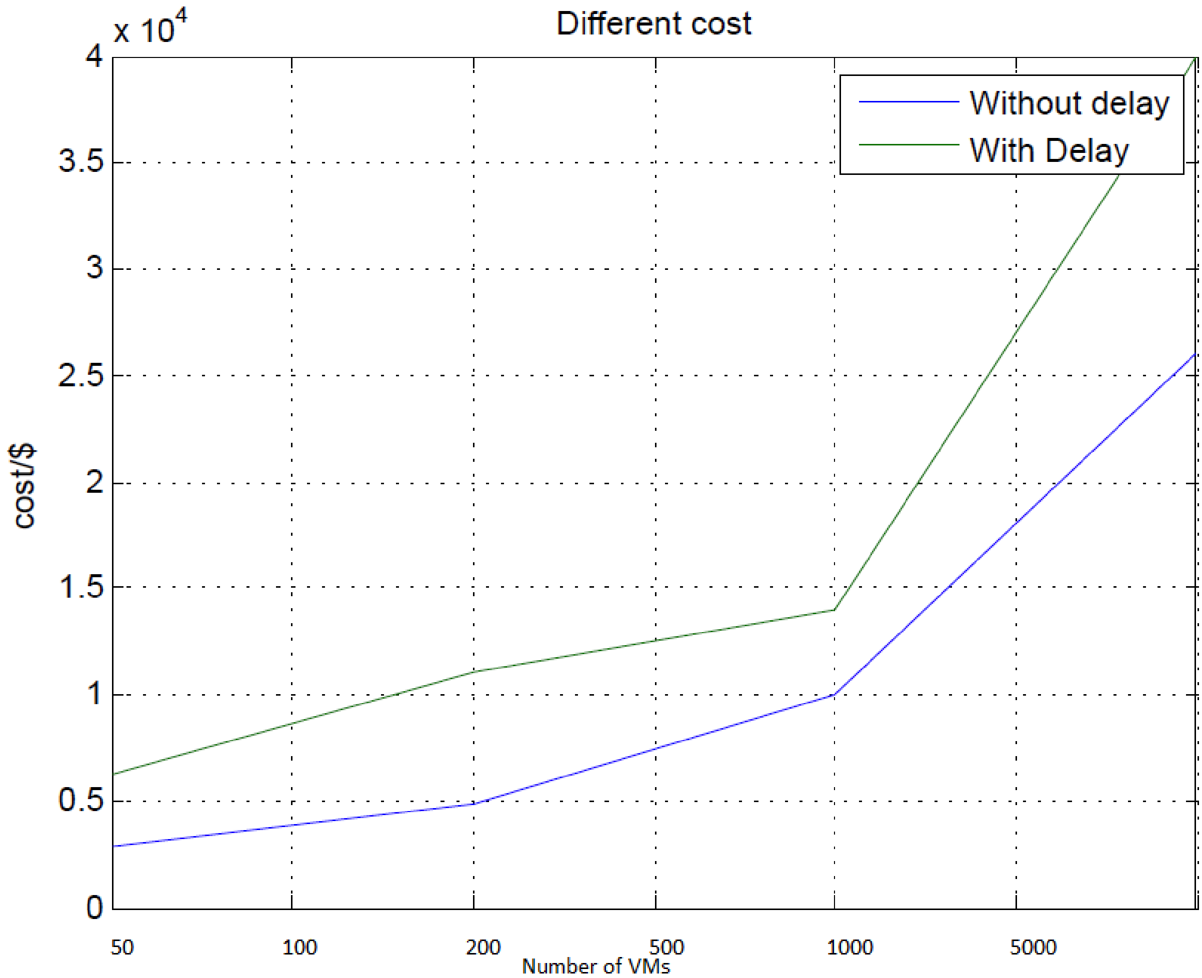

Figure 15: Comparison of cost

By increasing the number of nodes for the same load (VM requests) this result is obtained.

Figure 19 shows that with the same load the cost of finding optimal solution is decreased because it has more options to find solutions and can find a path with less cost.

7.1 Problem in running optimization

While running the optimization for large number of requests (5000 in this case), we encountered a problem. Suddenly, the running time became about four hours compared to two minute for 1000 request, and minimal cost got zero. This showed that some constraints cannot be satisfied.

7.2 Solving optimization problem

In order to solve the problem, it is assumed that each that data center has an extra node and a related link to it. This node has a big size (infinite) and also the cost of using this node (CPU cost or storage) is very high. Moreover, the same holds for the connection link between

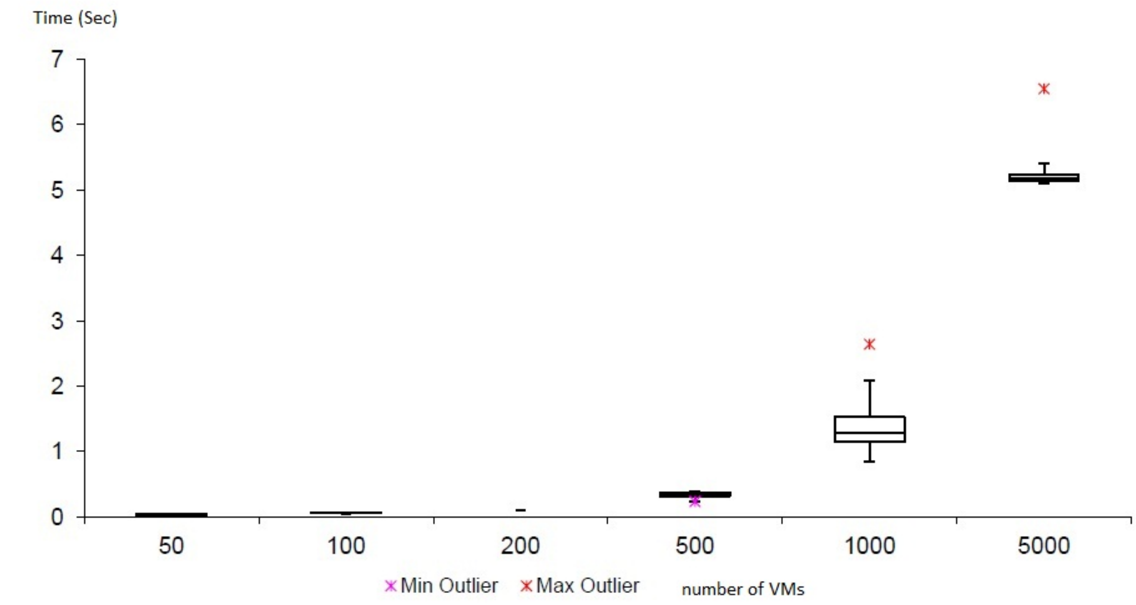

Figure 16: Time without delay

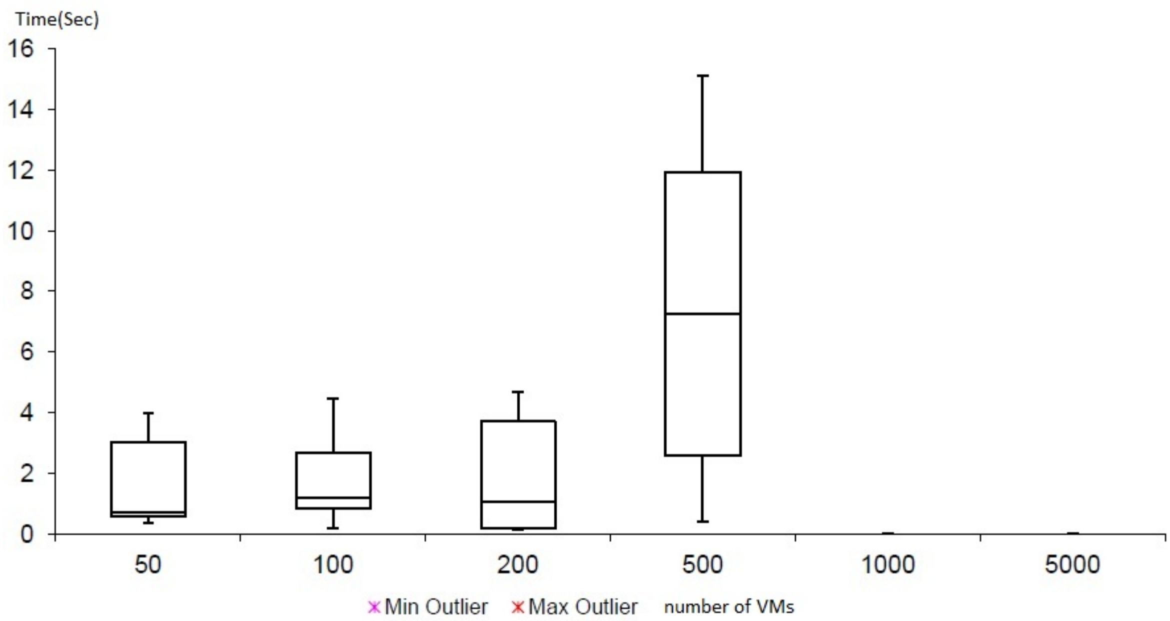

Figure 17: Time with delay

the extra node and the main node. Therefore, when the number of requests is high (high load) and there is not enough capacity ether in nodes or links, the requests will be allocated in these extra nodes. The result of the optimization after using this node is considerable.

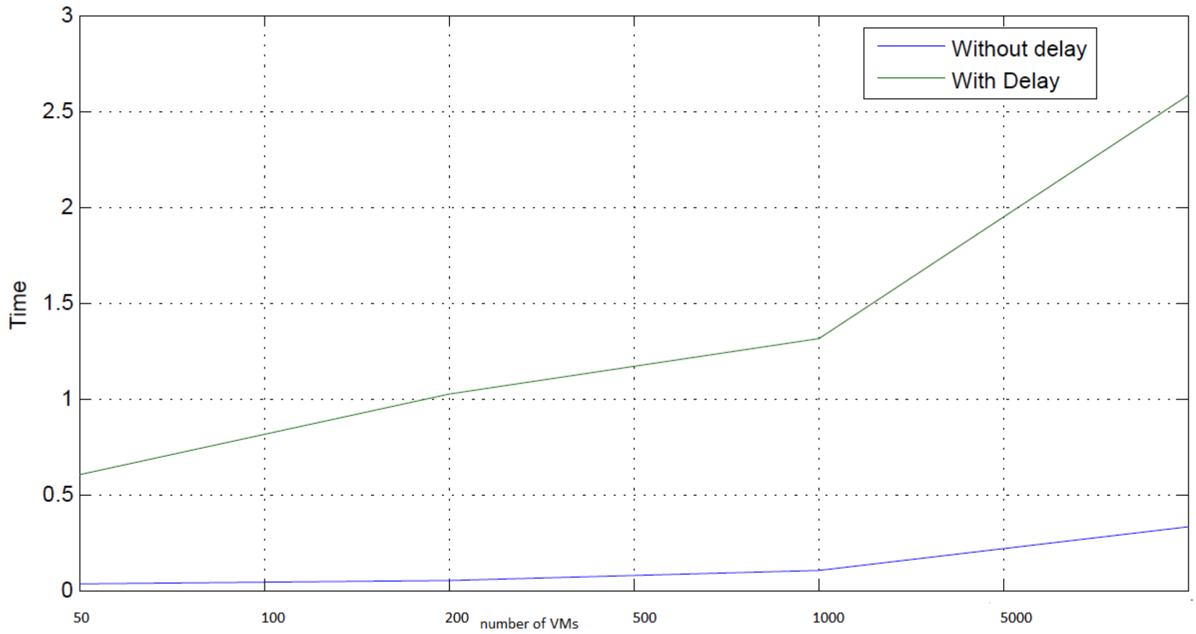

Figure 18: Comparison of time

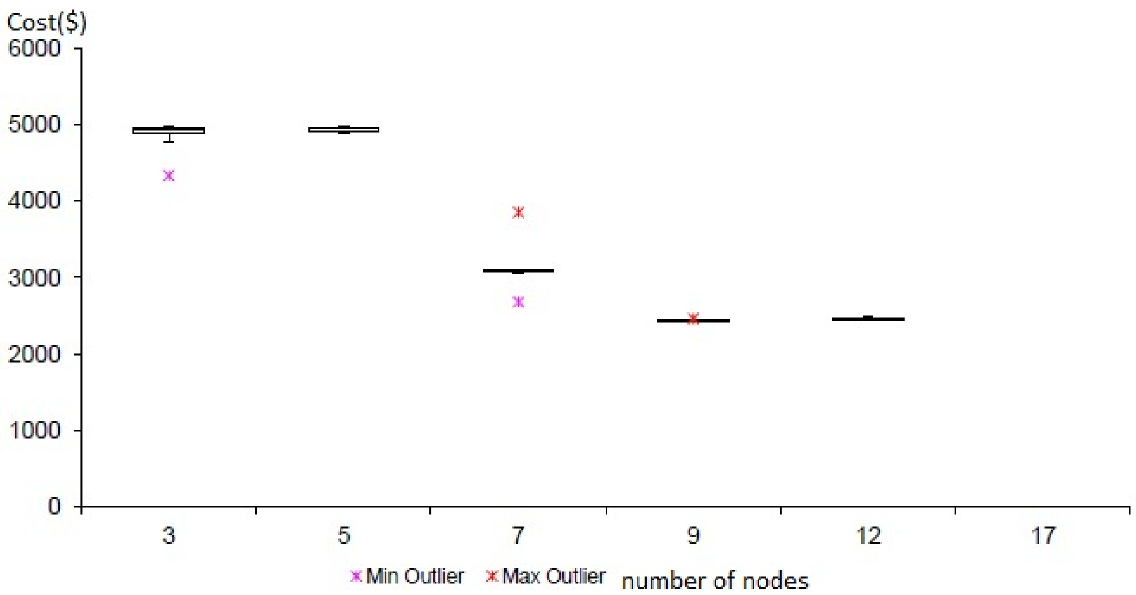

Figure 19: Increasing nodes

Figure 20 shows that the results suddenly get high meaning some constraints are not satisfied and some requests are allocated in extra nodes.

Figure 21 shows the modified Telenor network with extra nodes.

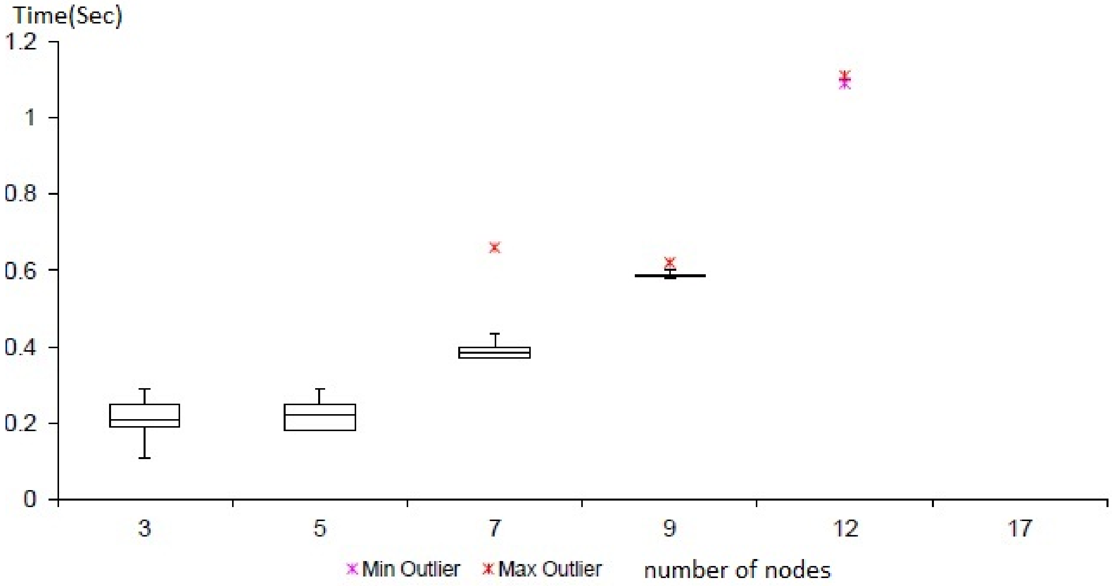

Figure 20: Increasing nodes

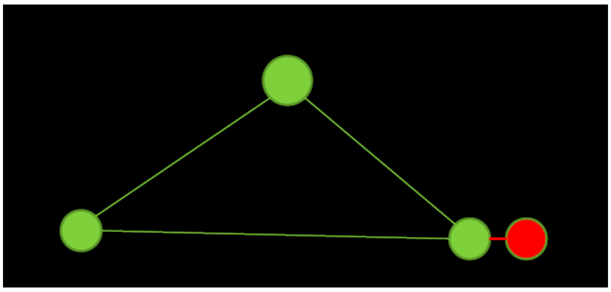

Figure 21: Modified topology

7.3 Future work

This project has been performed in Telenor. There is a potential to compare theoretical and practical results in the future. Another possibility is to present a new routing algorithm in order to get a better optimal results. (This can be done as the Master thesis) Moreover, the energy consumption in cloud network is an interesting issue. Therefore, it can be a good subject for future research.

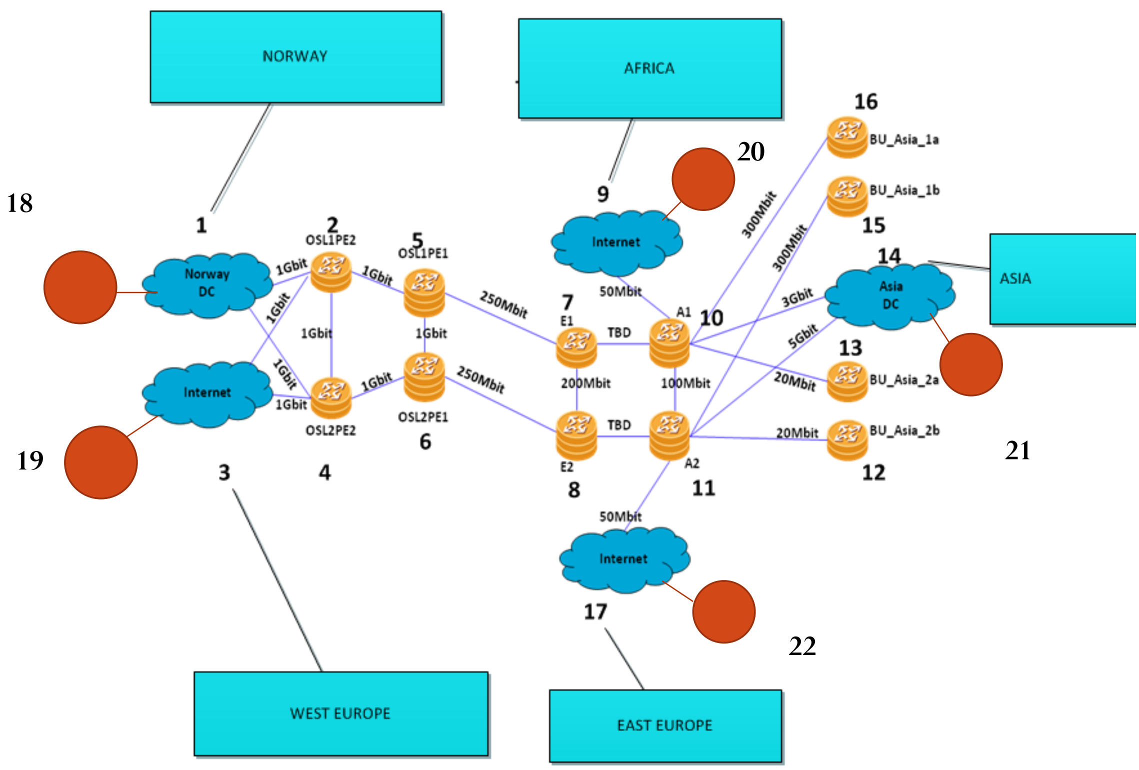

Figure 22: Modified Telenor network

8 List of Acronyms

SLA Service Level Agreement

PaaS Platform as a Service

IaaS Infrastructure as a Service

DaaS Data-storage as a Service

QoS Quality of Service

RAS Resource Allocation System

CPU Central processing unit

RDF Resource Description Framework

NDL Network Description Language

EDML Elastic Deployment Modeling Language

ECML Elastic Computing Modeling Language

EMML Elastic Management Modeling Language

ADL Application Descriptor Language

SML Service Modeling Language

SML-IF Service Modeling Language Interchange Format

VM Virtual Machine

DCA Datacenter Control Algorithm

OS Operation System

VMM Virtual Machine Monitor

RAM Random Access Memory

GB Gigabyte

GHz Gigahertz

LP Linear programming

Bps Bite Per Second

CEAS Cross Entropy Ant System

9 List of References

References

[1] A Break in the Clouds: Towards a Cloud Definition. Luis M. Vaquero1, Luis Rodero-Merino1, Juan Caceres1, Maik Lindner2. lmvg@tid.es, rodero@tid.es, caceres@tid.es, m.lindner@sap.com

[2] Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Buyya, Rajkumar, et al. 6, 2009, Future Generation Computer Systems, Vol. 25, pp. 599-616.

[3] Toward a Unified Ontology of Cloud Computing. Youseff, Lamia, Butrico, Maria and Da Silva, Dilma. 2008. Grid Computing Environments Workshop. pp. 1-10.

[4] Moving to cloud. Christian Mikalsen. Master thesis, 2009. University of Oslo department of informatics

[5] The NIST Definition of Cloud Computing. Mell, Peter and Grance, Tim. 2009, National Institute of Standards and Technology, Information Technology Laboratory.

[6] Resource Allocation in Clouds: Concepts, Tools and Research Challenges ,Glauco EstÃ¡cio GonÃ§alves, PatrÃcia Takako Endo, Thiago Damasceno ,Cordeiro, AndrÃ© Vitor de Almeida Palhares, Djamel Sadok, Judith Kelner,Bob Melander, Jan-Erik Mangs

[7] K. Krauter, R. Buyya, and M. Maheswaran, A taxonomy and survey of grid resource management systems for distributed computing. Softw.Pract. Exper. 32, 2 (Feb. 2002),

[8] R. Buyya and M. Murshed. (2002, 11). GridSim: A toolkit for the modeling and simulation of distributed esource management and scheduling for grid computing. Concurrency and Computation Practice and Experience 14(13-15), pp. 1175-220.

[9] Understanding Data Centers and Cloud Computing, Paul Stryer, Global Knowledge Instructor, CSI, CCNA,2010

[10] Virtualization Overview. VMware White Paper. 2006.

[11] Data center evolution: A tutorial on state of the art, issues, and challenges. Krishna,Kant. s.l. : Elsevier, 2009, Computer Networks

[12] N. Widell and C. Nyberg, 2004,"Cross Entropy based Module Allocation for Distributed Systems",Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing

[13] Swarm Intelligence Heuristics for Component Deployment Mate J. Csorba and Poul E. Heegaard Department of Telematics, Norwegian University of Science and Technology

[14] The Cost of a Cloud:Research Problems in Data Center Networks,Albert Greenberg, James Hamilton, David A. Maltz, Parveen Patel,Microsoft Research, Redmond, WA, USA

[15] Multi-objective Optimization Based Virtual Resource Allocation Strategy for Cloud Computing, Li Xu, Zhibin Zeng, Xiucai Ye

[16] Swarm Based Distributed Search in the AMIGOS Environment, Otto Wittner,Poul E. Heegaard, Bjarne E. Helvik

[17] AMPL at university of Michigan, D Holmes, 23 June 1992

[18] Haider A., Potter, R., Nakao, A. (2009) "Challenges in Resource Allocation in Network Virtualization". In 20th ITC Specialist Seminar, Hoi An, Vietnam, May.

[19] Ricci, R., Oppenheimer, D., Lepreau, J. and Vahdat, A. (2006) "Lessons from Resource Allocators for Large-Scale Multiuser Testbeds". In ACM SIGOPS Operating Systems Review, Vol. 40 , Issue 1, pp. 25-32, January.

[20] http://en.wikipedia.org/wiki/Virtual_machine

[21] Introduction to cloud computing ,William Voorsluys, James Broberg, and Rajkumar Buyya

10 Appendices