Guidelines for Building a Private Cloud Infrastructure

Zoran Pantić and Muhammad Ali Babar

Tech Report TR-2012-153

ISBN: 978-87-7949-254-7

IT University of Copenhagen, Denmark, 2012

Summary

Cloud computing has become an extremely attractive area of research and practice over the last few years. An increasing number of public and private sector organizations have either adopted cloud computing based solutions or are seriously considering a move to cloud computing. However, there are many concerns about adopting and using public cloud solutions. Hence, private cloud solutions are becoming an attractive alternative to a large number of companies. We initiated a project aimed at designing and setting up a private cloud infrastructure in an academic and scientific environment based on open source software. One of the key objectives of this project was to create relevant material for providing a reference guide on the use of open source software for designing and implementing a private cloud.

The primary focus on this document is to provide a brief background on different theoretical concepts of cloud computing and then elaborate on the practical aspects concerning the design, installation and implementation of a private cloud using open source solution. It is expected that organizations looking at the possibilities for implementing cloud solutions would benefit from getting the basics, and a view on the different aspects of cloud computing in this document. Defining the cloud computing; analysis of the economical, security, legality, privacy, confidentiality aspects. There is also a short discussion about the potential impact on the employee's future roles, and the challenges of migrating to cloud.

The main part of this report is concentrating on the practical infrastructure related questions and issues, supplied with practical guidelines and how-to-dos. The main topic was the design of the server and network infrastructure, and the distribution of the roles over the servers belonging to a private cloud. The management of the instances and the related subjects are out of the scope of this document. This document is accompanied by three supplemental books that contain material from our experiences of scaling out in the virtual environment and cloud implementation in a physical environment using the available server and networking equipments.

The conclusions drawn from the reported project are that the private cloud solutions are more customizable, but have greater installation cost, and lower flexibility. They are at an early stage, not much predictable, still too hard to install, manage and maintain for an administrator without an advanced knowledge and practical skills in different aspects of advanced of IT infrastructure solutions. There is a steep learning curve for the whole organization, for both users and administrators. Public clouds are more mature, well documented, rich with features, and easy to use. But the effort required for running a private cloud is having a downward tendency, and is expected to meet the level of effort required for implementing solutions on a public cloud at some point in time. For organizations not familiar with the technology used for private cloud implementations, a better choice would be going for a public cloud implementation, meanwhile learning and working with the technology, building the operational skills while waiting for it to be more mature.

Table of Contents

1. Introduction

Cloud Computing has recently emerged as an attractive model of providing Information Technology (IT) infrastructure (i.e., computing, storage, and network) to large as well as small enterprises both in private and public sectors. An enormous surge in the popularity of cloud computing is partly driven by its promise of on demand scalability and flexibility without making upfront investment in setting up and running large scale computing infrastructures and data centers. The proponents of cloud computing argue that it can enable organizations to scale up or scale down their acquisition of IT infrastructure based on their consumption patterns [1-2]. A user of Web applications may have been utilizing the platforms or applications deployed in a cloud computing infrastructure as it has many facets and forms. For example, most of the state-of-the-art social networking sites such as Facebook, Twitter, YouTube, and Flickr are supported by high-performance cloud platforms and application engines provided by some of the leading public cloud providers such Google and Amazon.

Apart from popular social networking sites, several hundred companies have started providing and using cloud-enabled services and thousands of others are seriously considering entering in this foray. However, there are several confusions about this emerging paradigm, which appears to have taken IT World (including computing and software engineering) practitioners in general and researchers in particular by surprise. One of the commonly observed confusion is about what cloud computing means as different people may talk about different things when using the word cloud computing. There have been several definitions of cloud computing [1, 3-4]. Like many practitioners and researchers, we also use the following definition of cloud computing provided by the US National institute of standards and technology (NIST).

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [4].

There are three main categories of service models of cloud computing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [2]. Scalability is one of the most prominent characteristics of all three categories [5-6]. The IaaS systems can offer elastic computing resources like Amazon Elastic Compute (EC2) and on demand storage resources like Amazon's Simple Storage Service (S3). Two of the most common deployment models of cloud computing are public cloud infrastructure and private cloud infrastructure. The former are the cloud computing infrastructure provided by 3rd party service provider (such as Google and Amazon) based on pay-as.-you-use model and the latter is the cloud computing infrastructure set up and managed by an organization for its own use.

Both kinds of deployment models have been gaining popularity for several reasons for example, lower cost of entry, reduced risk of IT infrastructure failure, higher ROI, quick responses to changes in demand, rapid deployment, increased security, and ability to focus on an organization' core business. However, there are certain legal, political, socio-organizational reasons that may discourage (or even bar) an organization from using public cloud infrastructure for certain kinds of activities, for example processing and storing citizens' private data. For these kinds of situations, private cloud infrastructure is considered an appropriate alternative. Hence, private cloud infrastructure is gaining much more popularity than the public cloud solutions in certain parts of the World, e.g., Europe.

Another big reason for an increased interest in setting up and managing private cloud in Europe is a significant amount of uncertainty about the potential EU regulation in terms of privacy, security, location and ownership of data. Moreover, a large number of private and public sector enterprises count heavily on the open source community (more than 50% open source community is based in Europe) that is why the US based cloud providers are facing resistance from local European governments [7]. UK, the Netherlands, and the Nordic countries are at the forefront of cloud enablement, and it seems that the local and regional clouds will dominate in the coming years [7].

On the economic side, the need for processing large data volumes are growing strongly in both industry and research, the need to conserve power by optimizing server utilization is also booming. Moreover, there is an increasing demand for consolidating IT resources due to economic conjuncture (recession). The market is on its way to on-demand computing power for both small businesses and research, which are increasingly exploring the option of having private cloud solutions, especially the academic and scientific sectors which have large amount of students' and research data that may not be rendered to public cloud providers' infrastructures.

Despite an increasing interest in public cloud computing solutions, there is hardly any comprehensive guide to build, operating, trouble-shoot, and manage a private cloud infrastructure. The acute shortage or unavailability of guidelines for building and managing a private cloud using an Open Source Software (OSS) is being increasingly felt in academic and scientific Worlds. This paper reports a preliminary set of guidelines for building a private cloud infrastructure in an academic environment using OSS like Eucalyptus and Open Nebula. These guidelines have been derived from our practical experiences from successfully completing a project on building a private cloud infrastructure for research and education using OSS. The main focus of this project was on practical design and implementation rather than theoretical approaches to building private cloud infrastructure. One of the key goals of the project was to produce tutorial style guidelines for designing and implementing private cloud solution. Possibilities for implementing in Scientific Computing were also researched [8]. We believe that one of the options available to Danish academic and research sectors is to build, operate, and maintain their own private clouds. Hence, these guidelines can be valuable to the organizations interested in building their own private cloud infrastructure.

2. Background of Cloud Computing

2.1 Emergence and Popularity of Cloud Computing

Cloud computing has emerged as a popular paradigm of acquiring and providing computing, network, and storage infrastructure that promises on demand elasticity for scaling up and/or scaling out. That essentially means off-loading parts of organization's server room to highly automatized commercial data centers. Apart from enormous development in the supporting technologies like virtualization and utilization, another key reason for the rapid uptake is the backing of reputable vendors like Google, Yahoo and Amazon, which are known for having reliable infrastructures, who started opening their datacenters for public use. In a laymen's term, cloud computing is a highly automatized datacenter that has Web-based access to computing, storage, and networking service which are available on-demand without any need of human intervention; and where use and billing pattern are described as "use-only-what-you-need" and "pay-only-what-you-used".

The foundations of the key concepts unpinning cloud computing date back to 1966 when Douglas Parkhill [9] wrote his book "The Challenge of the Computer Utility"[1]. Parkhill compared his ideas about utility computing with the electricity industry. Nicholas Carr compared it to the revolutionary change from using steam engines to electrical energy delivered by the big, central electric companies [10]. Now we have been experiencing the changes in the provision and utilization of IT infrastructures based on the ideas presented many years ago but the underlying technologies were not available. Whilst cloud computing is still in its infancy stage of adoption and apparently suffers from several kinds of challenges and threats, the rapid development of supporting technologies and systems bode well for the future of this phenomenal model of acquiring IT infrastructure.

According to Gartner's "Hype Cycle for Emerging Technologies" of August 2010, Figure 1 shows that Cloud Computing has reached its peak in expectations, being on its way to a more mature state, and with time it is expected to reach the stability that it needs so much for stable and steady growth in productivity.

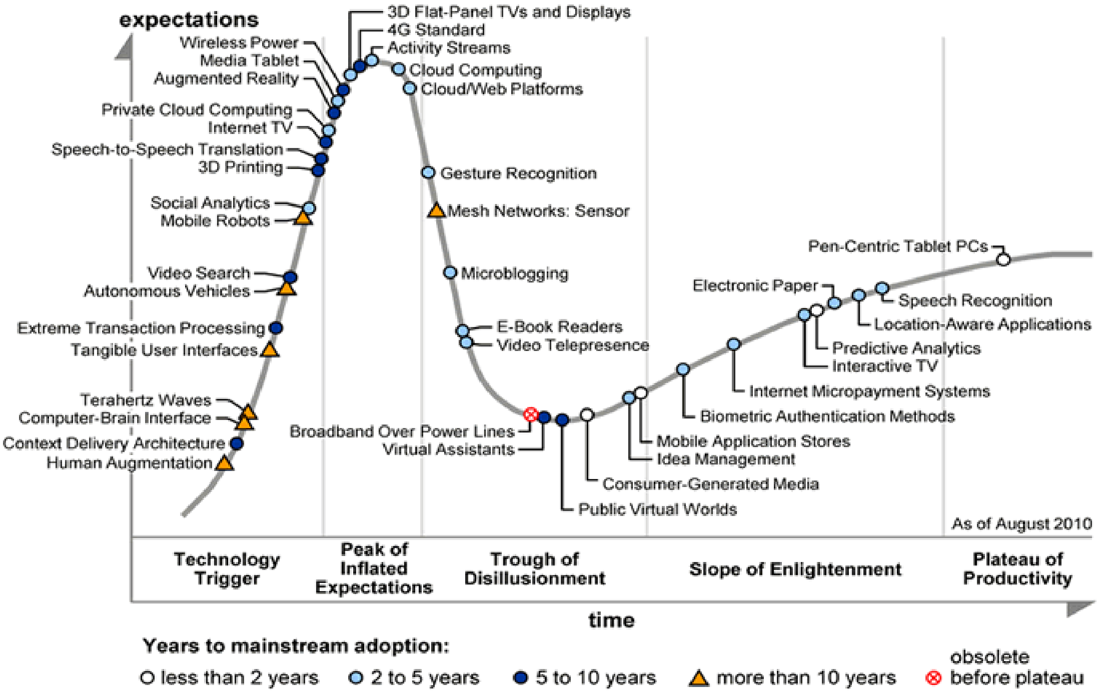


Figure 1 - Source: Gartner (August 2010)[2]

[1] http://en.wikipedia.org/wiki/Douglas_Parkhill

[2] http://blogs.gartner.com/hypecyclebook/2010/09/07/2010-emerging-technologies-hype-cycle-is-here

When we review the Gartner's "Hype Cycle for Cloud Computing" of July 2010 as shown in Figure 2, it is clear that private cloud computing still didn't reach the hype-peak; rather it had a long way to go. This was also confirmed by our interview study carried out to identify the needs and requirements of building a private cloud infrastructure in an academic and scientific environment. There have also been several reports in the trade media that the private cloud technology is still not mature enough.

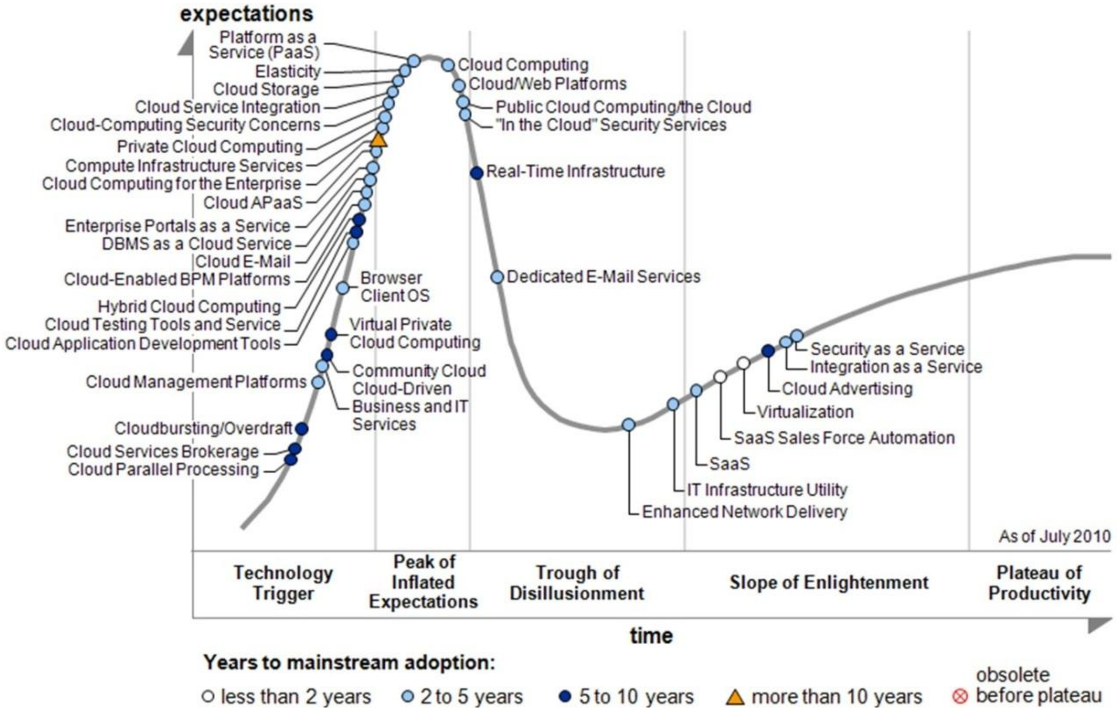


Figure 2 - Source: Gartner 2010 / Computerworld 10/2011[3]

However, it is becoming increasingly obvious that the hype-curve of private cloud will be fast forwarded for several reasons. One of the key reasons is the stringent legal requirements of data movement and protection in many European countries. That is why there are many countries (including Denmark) which characterize public cloud computing whose security and privacy measures for data protection are yet to be proven. Many organizations like the Council for greater IT-security in Denmark state that the only safe way to use cloud computing is to implement some sort of private cloud[4] in order to have the direct ownership and responsibility for the physical infrastructure, stability, IT-security and protection of data privacy. There also several cases where public departments have been discouraged or stopped from using public cloud computing infrastructure. For example, Danish IT-security bodies (Datatilsynet and IT-sikkerhedsråd) advised against the use of cloud solutions in the public sector. For example, Odense Kommune was banned from using Google Apps[5], and Kommunernes Landforening's driving-test booking system was not allowed to be implemented using Microsoft's Azure cloud[6]. Not being able to use public cloud solutions, an increasing number of private and public sector companies are tuning towards private cloud based solutions.

[3] http://www.computerworld.dk/art/116902/her-er-de-mest-hypede-cloud-teknologier
[4] http://www.version2.dk/artikel/raadet-it-sikkerhed-offentligt-cloud-loesninger-er-usikre-29166
[5] http://www.computerworld.dk/art/113925
[6] http://www.computerworld.dk/art/115728

2.2 Essential Characteristics of Cloud Computing

An apparently meteoric rise in the popularity of cloud computing has several technical and economic reasons. Some of these reasons are considered the essential characteristics of cloud computing based solutions. In the following, we enlist some of the most frequently reported characteristics of cloud computing [1]. That means a cloud computing based solution, private or public, is expected to demonstrate the following characteristics.

- On-demand self-service – a consumer of cloud computing solution should be able to automatically acquire and release the IT resources without requiring any action from the service providers whenever the need for such resources increases or decreases;
- Broad network access – the cloud computing based IT resources are available over the network and accessed by thin or thick client platforms;
- Resource pooling – the available computing resources, physical or virtual, are pooled together and are dynamically assigned and reassigned based consumers' demand; multiple consumers are served using a multi-tenant model;
- Rapid elasticity – the provisioning and releasing of the resources, instantly and elastically, are preferably done in an automatic way in order to enable a consumer to quick scale out and in; compared to customer's demand, the resources may appear to be unlimited, available in any quantity at any time;
- Measured Service – a cloud computing solution also has the ability to measure the consumption of resources, and to automatically control and optimize the resources; this happens on some level of abstraction corresponding to the type of service; the consumption can be monitored, controlled ("hard quotas"), and reported ("soft quotas").

2.3 Services and Deployment Models of Cloud Computing

One of the main reasons for a meteoric rise in the popularity and adoption of cloud computing is its enormous level of flexibility for scaling up or scaling down software and hardware infrastructures without huge upfront investments. Hence, it is expected that any cloud-based infrastructure should have three characteristics: ability to acquire transactional resource on demand, resource publication through a single provider, and mechanisms to bill users on the basis of resource utilization [6]. Cloud computing provisions and acquisitions are usually characterized by well known services and deployment models. In order to fully understand and appreciate the need and value of having a private cloud, we briefly describe the commonly known services and deployment models of cloud computing in this Section. Figure 3 and 4 presents the pictorial views of the most commonly known service and deployment models of cloud computing. For a details discussion about the service and deployment models of cloud computing, see these publications [1-2, 6, 11-14].

2.3.1 Service Models

A service is defined as a fine-grained reusable resources (i.e., infrastructure or business processes) available from a service provider; this is now what is popularly called "as a service". Services should have low entrance barriers (making them available for low budget enterprises), large scalability, multitenancy (same resources shared for many user, without interference), and support for variety of access possibilities concerning different types of hardware infrastructures and operating systems.

- Software as a Service (SaaS) is a kind of application that is available as a service to users; it delivers software as a service over the Internet, eliminating the need to install and run the application on local computers in order to simplify the maintenance and support. One of the main differences of using such an application that the application is usually used without being able to make a lot of adaptations and preferably without tight integration with other systems. Such applications are accessible from various client devices using a thin client interface such as a web browser; the service provider does all the operation and maintenance from the application layer to the infrastructure layer. The applications that are good candidates for being offered as SaaS are accounting, video conferencing, Customer Relation Management (CRM) and IT Service Management. One of the benefits of SaaS is lower cost, user familiarity with WWW, and web availability and reliability;

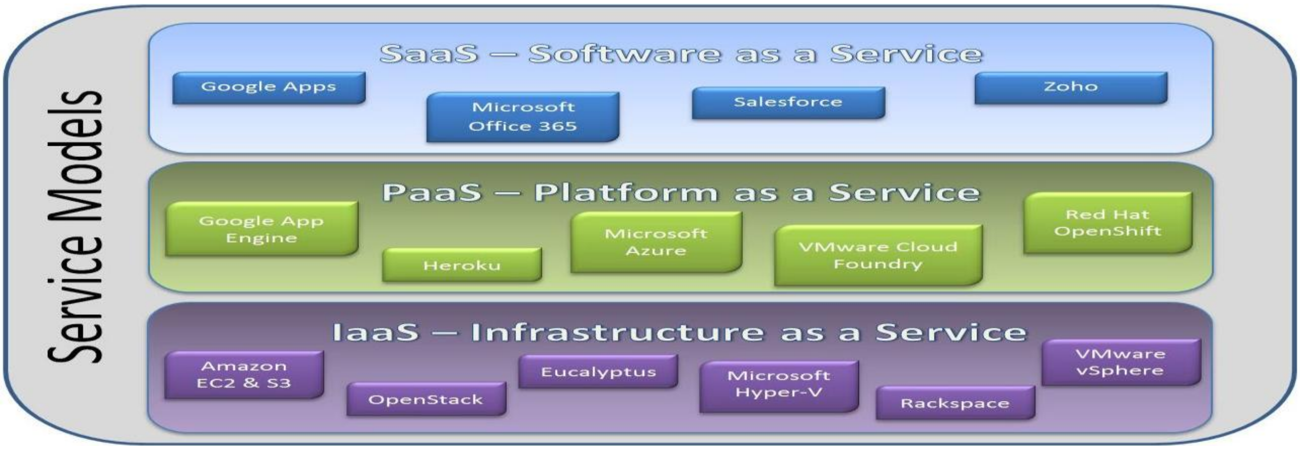


Figure 3: Most commonly known service models of cloud computing

- Platform as a Service (PaaS) supplies all that is needed to build applications directly from the Internet with no local software installation. The PaaS service model enables the deployment of applications without the cost and complexity of buying and managing the underlying hardware and software layers. A customer can deploy an application directly on the cloud infrastructure (without managing and controlling that infrastructure) using the programming languages and tools supported by a provider. A customer has the control over its applications and hosting environment's configurations. It supports web development interfaces such as Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). The interfaces allow for the construction of multiple web services (mash-ups), and are also able to access databases and reuse services available within private network;
- Infrastructure as a Service (IaaS) delivers a computer infrastructure that is a fundamental resource like processing power, storage capacity and network to customers; instead of building data centers, purchasing servers, software or network equipments, a customer buys the resources as a fully outsourced service; a customer does not manage the underlying infrastructure but has full control over the operating systems and the applications running on it. IaaS models often provide automatic support for on demand scalability of computing and storage resources.

2.3.2 Deployment models

Following are the most commonly known deployment models as shown in Figure 4.

- Private cloud refers to a cloud infrastructure that is internal to an organization and is usually not available to the general public. A private cloud infrastructure is normally operated and managed by the organization that owns it. However, an organization can also get a public cloud provider to build, operate, and manage a private cloud infrastructure for an organization. Sometimes the operation and/or management of a private cloud infrastructure can also be outsourced to a 3rd party. A private cloud's data centers can be on premise or off premise (for security and performance reasons);
- Public cloud is a cloud computing infrastructure that is made available as "pay-as-you-go" and accessible to the general public. A public cloud's service is sold along the lines of the

concept called utility computing[7]. A public cloud infrastructure is owned by an organization that is providing the cloud services such as Amazon Web Services and Google AppEngine;

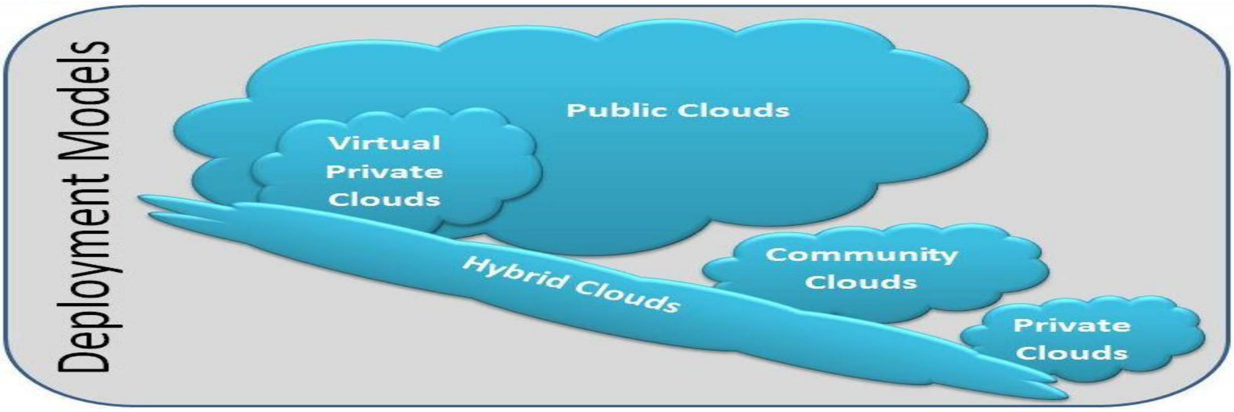


Figure 4: Most commonly known deployment models of cloud computing

- Community cloud refers to a cloud computing infrastructure set up by a consortium of several organizations that are likely to have common requirements for a cloud computing infrastructure; hence, they can agree on implementing a common infrastructure that is reliable and cost effective. The operation and/or management of a private cloud infrastructure are the responsibility of the consortium or they can decide to outsource those responsibilities to a 3rd party. A community cloud's data centers can be on or off premise (for security and performance reasons).
- Virtual private cloud is a cloud computing infrastructure that runs on top of a public cloud that has a customized network and security settings for a particular organization. Virtual private cloud is an attractive solution for organizations which wants to use cloud computing technology without incurring the cost of building and managing its own private cloud and also wants to avoid the security and data privacy risks that characterize public cloud services.
- Hybrid cloud deployment model is a composition of two or more cloud deployment models. The elements in a hybrid cloud composition remain unique but those are bound together by standardized or proprietary technology that enables portability of data and applications.

2.4 Motivation for Migrating to Cloud Computing

There can be several kinds of reasons for moving IT infrastructure to Cloud computing technology; for example, lower cost of entry, reduced risk of IT infrastructure failure, higher ROI, quick responses to changes in demand, rapid deployment, increased security, and ability to focus on an organization' core business.

Whilst there is uncertainty about the potential EU regulation in terms of privacy, security, location and ownership of data, cloud computing is on the EU digital agenda. EU is counting heavily on the open source community (more than 50% open source community is based in Europe), and the American cloud providers are facing resistance from local European governments [7]. UK, the Netherlands, and the Nordic countries are at the forefront of cloud enablement, and it seems that the local and regional clouds will dominate in the coming years [7].

On the economic side, the need for processing large data volumes are growing strongly in both industry and research, the need to conserve power by optimizing server utilization is also booming. Moreover, there is an increasing demand for consolidating IT resources due to economic conjuncture (recession). The market is on its way to on-demand computing power for both small businesses and

[7] http://en.wikipedia.org/wiki/Utility_computing

research, which are increasingly exploring the option of having private cloud solutions. Organizations are also looking at mobile applications[8] as obvious candidates for implementing on cloud computing [15]. Of course, that would not be possible if data were not available on-line, i.e., raw data and geodata[9], using Open Data Protocol[10]. Organizations willing to streamline their office applications implementation are looking at Google Docs and Office 365 [16].

It is quite obvious question for many organizations that what approach to take for migrating to cloud computing? It seems that a “big bang" approach is not a feasible approach to migrating to cloud, rather the migration should be done “in piecemeal” manner [7]. Such a migration can start with the data of low importance first, see how it goes, and then gradually move on, migrating the data of low importance. A large organization can apply ITIL[11] and COBIT[12] principles.

2.5 Economic Considerations

For the organizations whose core business is not information technology, the maintenance of an IT infrastructure is quite expensive. Recruiting suitably skilled staff and arranging their training and certifications, software licenses and hardware infrastructure need huge upfront investment and continuous maintenance and management responsibilities. It is quite understand if an organization wants to reduce the costs of its IT infrastructure and release the resources for allocating them to their core business needs. Outsourcing of IT infrastructure is one the strategies that many organizations use for reducing their costs and minimizing their responsibilities for setting up, upgrading, operating, and managing IT infrastructures. This strategy is expected to release the employees to fully concentrate on the core business activities for building and strengthening organizational strengths. Moreover, it also enables an organization to become more agile and adaptable for dealing with the modern economic turbulences [17].

Administration costs are also one area where cloud computing has its benefits, however, such cost reductions are pretty hard to estimate due to the fact, among other things, that administrators have many varying duties that could prevent them from concentrating only on the cloud computing solution. Whilst private cloud solution implementation and management would still be lower compared to the old fashioned computer center, private cloud solutions are more expensive to implement and manage compared to public cloud solutions.

Another aspect of the cost involved in setting up and running a private cloud is the total cost of ownership (TCO), which not only involves the initial set up cost but also electricity cost, the cost of renewing the hardware, and staffing costs. However, a private cloud in the Nordic region[13] not only need relatively less amount of energy for cooling data centers because of the geographical latitude but can also be powered through a number of alternative energy sources for example thermal energy sources on Island, water power plants in Norway, Sweden and Finland.

2.6 Legal and Regulatory Considerations

One of down sides of the cloud computing is the potentially negative impact on privacy and confidentiality. For certain kinds of applications, legal regulations may not allow the use of cloud computing services provided by public cloud providers. Hence, there has to be a private cloud computing solution for the applications that deal with private data such as personal identification numbers (CPR in Denmark) or medical records of Danish citizens.

Since Cloud computing services are accessed via Internet, it is reasonable to expect that these services are easier to access and more exposed to potential attacks compared to the closed networks

[8] https://datamarket.azure.com

[9] http://www.data.gov

[10] http://www.odata.org

[11] http://en.wikipedia.org/wiki/ITIL

[12] http://en.wikipedia.org/wiki/COBIT

[13] http://www.norden.org/en/news-and-events/news/joint-nordic-supercomputer-in-iceland

and systems within an organization's security perimeters. There are, of course, solutions to the potential risks associated with cloud computing, for example, encryption and splitting the data to several places, however, these kinds of solutions may not necessarily address a large number of potential risks that legal authorities will like organizations to take into considerations with regards to the security and privacy of the data they deal with. A private cloud infrastructure is being promoted as a solution that can enable an organization to maintain a tight control on its IT infrastructure according to its legal responsibilities while benefiting from the cloud computing technologies.

2.7 Socio-Technical Considerations

Cloud computing can be considered a competitive solution for implementing in scientific and high performance computing (HPC) environments, but only in low end, for low memory and low core number jobs (single threaded jobs with small memory and I/O requirements). According to an estimate, one third of the scientific computing tasks could be ported to private cloud implementations[14] so that an HPC infrastructure can be used for the HPC processing tasks. For implementing a private cloud for scientific computing, there can be several challenges that can be clustered in two groups: sociological and technological.

Sociological challenges are mainly of political and economic reasons:

- Existing structures would oppose implementation of private cloud,
- There is a weak transparency of who is in charge of systems and economy,
- Research cannot be market cost-effective,
- People in charge are de facto administrators, instead of scientific groups,
- Tendency of IT department implementing things because they are interesting and "fun", while maybe there is no need for those systems.

Technological challenges:

- Private cloud maturity,
- Problems porting of programming code,
- IT Departments should be big enough, with enough expertize to be able to run private cloud for scientific purpose, since i.e. different batch jobs that need to be run cannot easily be standardized because of variety of scientific projects, and because researchers normally need advanced research instruments.

Based on all that, here is a list of suggestions that would increase the chances for success of implementing a private cloud:

- To determine the needs and their nature, consult the professors that are in charge of the project (and its funding),
- Once started, implementation should be top-down steered,
- A test case should be designed and implemented,
- Researchers should be allowed to thoroughly test the solution free of charge,
- Make sure that implementation succeeds first time.

The overall suggestion when it comes to implementing private cloud in scientific computing would be to gain a very clear picture of what services are to be offered, who will use them, what they will use them for, and how. Afterwards, take those findings in concern, together with the above mentioned challenges, and then make the decision.

[14] Rene Belsø, see Appendix 1

It is worth mentioning that Amazon's initiatives for academic use can also be explored for a general purpose cloud implementation. One of these offerings are the Amazon Education program with grants for research applications. Having a project, an academic organization can apply for a recurring grant, gets the approval within two weeks' time [18], and starts using it immediately afterwards. Locally in Denmark, CABO[15] was willing to supply the project with resources. These can be considered feasible alternatives to building a private cloud if the budget is low and the project should start as soon as possible.

3. Designing and Implementing a Private Cloud Infrastructure

3.1 Designing and choosing the solution

The very first step in the designing process is to get involved as many of organization's stakeholders as possible. One person cannot know everything. A solution that seems to be just right for the organization might have hidden issues, like security or privacy obstacles, that one is not aware of. In that case, knowing about the existence of the obstacles, it would have impact on the design decision of choosing private or public cloud solution. Such a wrong decision in the early stages of the project could have a great economic impact later in the project, not to talk about the possible privacy and security breaches.

Of course, the benefits of provisioning the servers needed in just a few clicks are enormous. But, one should also think about lock-out situations, where one is depending on one provider. Solution should be designed to include two or several providers, and standardizing is helpful in obtaining some sort of independence. Moving an organization's data to a cloud provider brings the issue of exporting the data, and all the providers are more than happy to supply the organization with the tool that can perform the export. However, it is equally important to consider the measures required in order to get the organization's data back again for in-sourcing or just changing a storage provider.

3.2 Why Ubuntu Enterprise Cloud based on Eucalyptus?

Ubuntu private cloud with Eucalyptus is chosen for the implementation for several reasons. For the first, it is Open Source Software (OSS), which was one of the requirements for the project. In academic and educational environments, OSS solutions are mainly used, for the sake of "interchangeability" between different organization (instead of being limited by using different proprietary software), and because of socio-organizational reasons. It is a common observation that researchers and scientists in academia usually do not prefer proprietary solutions if they can find something equally good from open source community. Ubuntu Enterprise Cloud (UEC), which is Ubuntu Linux distribution with (among other) Eucalyptus cloud software incorporated, is suitable since it is designed and suited to support private cloud solutions from a small starting point, and then scaling up/out as needed. This bundling of the Ubuntu Server Edition with Eucalyptus and other applications makes it very easy to install and configure the cloud. Eucalyptus has been designed as modular set of 5 simple elements, that can be easily scaled [9].

Another advantage is that UEC is using the same API's as Amazon [19], which makes " cloud bursting" or a hybrid cloud solution easily implementable. Moreover, if an organization has an existing virtualization infrastructure based on VMware, a Eucalyptus based solution can seamlessly overlay on top of an existing ESX/ESXi environment by transforming VMware based hypervisor environment into an Amazon AWS-like on-premise private cloud.

Amazon Web Services (AWS)[16] is one of the major players providing Infrastructure as a Service (IaaS), which is a reason for UEC leaning on it for scaling. There are many products and services

[15] Correspondence with founder Christian Orellana, CABO: http://www.cabo.dk/om-cabo-en

[16] http://aws.amazon.com

available under AWS, like the two services for respectively computing and storage: Elastic Compute Cloud (EC2)[17] and Simple Storage Service (S3)[18]. Those services are available through web services interfaces, and client tools can communicate with those services using EC2 and S3 APIs. The popularity of these APIs made them de facto standard that naturally leads to other cloud products' support for them. Eucalyptus is a cloud computing platform (available under GPL) which is EC2 and S3 compatible, and since its services can be reached using EC2 and S3 compatible APIs, any client tools written for AWS can communicate with Eucalyptus as well.

Eucalyptus offers both computing and storage, while others (like i.e. Open Nebula), as for the time being, are offering only computational power. Ubuntu is the fastest growing Linux-based operating system with over 10 million users and a huge community of developers [20-21]. Canonical, a company that supports and sponsors Ubuntu, specializes in the free provision of software for common activities. Canonical will also be offering commercial support for internal clouds, along with technical consultancy for installation. Hence, a widespread community support as well as commercial support is available for Eucalyptus. Refer to UEC/Glossary for explanation of Ubuntu Enterprise Cloud related terms used in this document[19].

3.3 Major UEC/Eucalyptus components

UEC/Eucalyptus is an on-premise private cloud platform, designed as a distributed system - a modular set of 5 simple elements. Linux based processes that interact with each other form the cloud layer. The components (Figure 6) that comprise the Eucalyptus architecture (Figure 5) are:

1. Node Controller (NC)
2. Cluster Controller (CC)
3. Walrus Storage Controller (WS3)
4. Storage Controller (SC)
5. Cloud Controller (CLC)

Each element is acting as an independent web service that exposes Web Service Description Language (WSDL) document defining the API to interact with it.

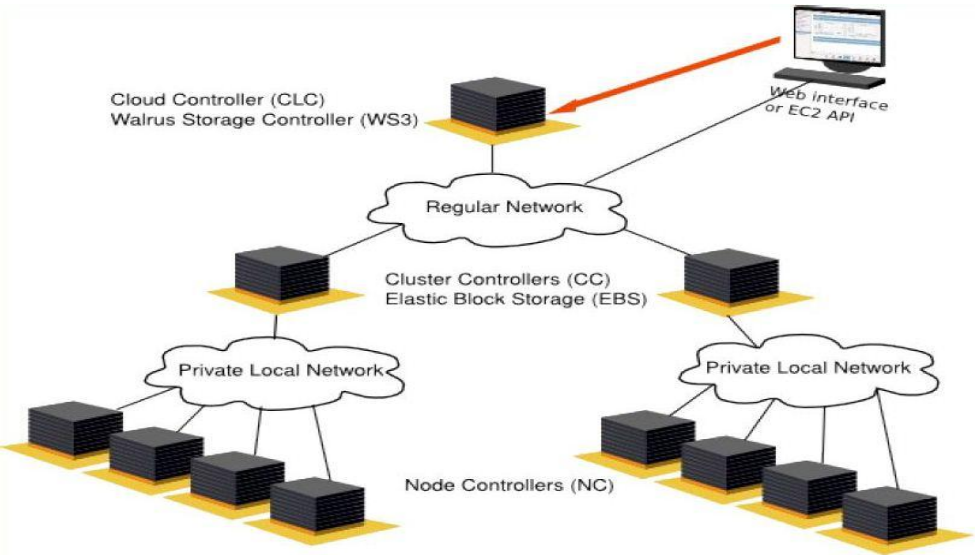


Figure 5 - Architecture of Eucalyptus [9].

[17] http://aws.amazon.com/ec2
[18] http://aws.amazon.com/s3
[19] https://help.ubuntu.com/community/UEC/Glossary

3.3.1 Node Controller (NC)

Compute nodes in the cloud ("work-horses").
Eucalyptus supports several hypervisors: KVM and Xen in open source version, and VMware (ESX/ESXi) in Enterprise Edition. Kernel-based Virtual Machine (KVM) is a virtualization solution for Linux on hardware that contains virtualization extensions. In Eucalyptus, KVM is chosen as preferred hypervisor. VT enabled server capable of running KVM is having Intel VT or AMD-V virtualization extensions. Allowing multiple operating systems to run concurrently as virtual machines on a host computer, hypervisor manages the execution of the operating systems of the virtual machines, called "instances". The NC is running on each UEC node and is controlling the instances on the node. NC is communicating with both the operating system and the hypervisor running on the node, and Cluster Controller. It gathers the data about physical resource availability on the node and their utilization, and the data about instances running on that node, and reports the data to the Cluster Controller.

3.3.2 Cluster Controller (CC)

A cluster is a collection of machines grouped together in the same network broadcast domain. CC is the entry point to a cluster. It manages NCs and instances running on them. It is communicating with Cloud Controller and Node Controller, receiving requests for deploying instances from CLS, and deciding about the NCs they will be used for deploying them. It also controls the virtual network available to the instances, and collects information on NCs, reporting it to CLC.

3.3.3 Walrus Storage Controller (WS3)

WS3 is equivalent to Amazon's S3. It is implementing the S3-like functionality: bucket based storage system with put/get storage model (create and delete buckets, create objects, and put or get those objects from buckets).
WS3 is storing the machine images and snapshots. It is a persistent simple storage service (using S3 API compatible REST and SOAP APIs) storing and serving files.

3.3.4 Storage Controller (SC)

SC is providing the persistent storage for instances on the cluster level, in form of block level storage volumes. The instances are using ATA over Ethernet (AoE) or Internet SCSI (iSCSI) protocol to mount the virtual storage devices. SC allows creation of block storage similar to Amazon's Elastic Block Storage (EBS)[20], and creation of storage volumes, attaching, detaching and creation of snapshots.

3.3.5 Cloud Controller (CLC)

CLC is the entry point to Eucalyptus cloud, and is equivalent to Amazon's EC2. It gathers information on the usage and availability of the resources in the cloud. It is the frontend for managing the entire UEC infrastructure. It has both web interfaces for administrators managing an infrastructure (such as images, users, groups, storage, network, and clusters), and the web

[20] http://aws.amazon.com/ebs

services interface (EC2/S3 compliant) for end users (client tools). It monitors the running instances, and the availability of resources on components of the cloud infrastructure. Based on the information on the infrastructure's load and resource availability, it arbitrates the available resources, dispatching the load to the clusters.

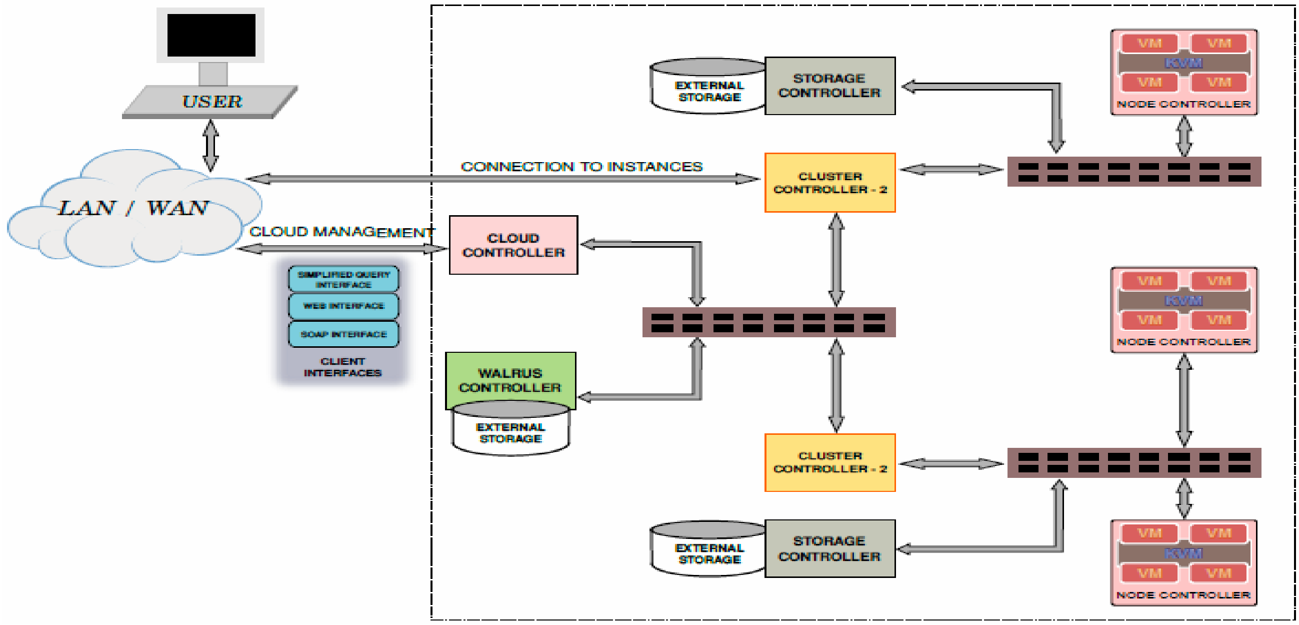


Figure 6 - Components of UEC [19].

3.4 Dimensioning the hardware infrastructure

Production UEC can consist of as few as two physical systems, up to potentially hundreds and thousands. For evaluation and experimental, it can also be installed on only one physical system, though with some limitations, which have been described a little later.

Based on the number and type of physical systems that are available in a cloud solution, there can be several topological recommendations depending on the needs and motivation for implementing the private cloud. Of course, the question of finding an ideal configuration is not only related to the needs, but also to available the budget. A nice thing about UEC architecture is its scalability: adding as many NC nodes at any time to any topology chosen is supported. So for answering the questions about ideal or recommended configuration and system architecture, how many servers are needed, how big the storage needs to be – one can take a very practical approach of implementing the solution on the hardware systems that are available at the moment, and then scaling up and out at the time a need for more resources arises. Some of the hardware suppliers have a "starter kit" solutions, dimensioned for certain number of virtual instances supported, but a more ad-hoc approach is also a good starting point, bearing in mind UECs scalability.

Concerning the hardware dimensioning[21], there are no fixed requirements, but rather recommendations. We can divide server types in two: a "front end & controller" type (hosting CLC/WS3/CC/SC components), and "NC" type (hosting NC component). Division is done based on the different load that the different components impose on the hardware: "front end & controller" type is not processor and memory demanding as the "NC" type is. Furthermore, NC type should have VT extensions enabled.

[21] https://help.ubuntu.com/community/UEC/CDInstall

Table 1. Suggestions for the “Frontend and controller” configurations

Hardware	Minimum	Suggested	Remarks
CPU	1 GHz	2 x 2 GHz	Depending on how many components are hosted on each physical server; if having all 4 components, at least dual core processor is recommended
RAM	2 GB	4 GB	Depending on the number of components
Hard disk	5400 rpm IDE	7200 rpm SATA	Slower/cheaper disk would work, but it is suggested to go for a faster versions (SATA, SCSI or SAN if the budget allows)
Disk space	40 GB	200 GB	40 GB is enough only for having a single image, cache, etc.; please dimension this with care, since Eucalyptus is sensitive to running out of disk space
Network	100 Mbps	1 Gbps	Copying machine images (hundreds of MBs) is done over network; if the budget allows, consider 10 Gbps)

Table 2: Suggestions for an “NC” configurations

Hardware	Minimum	Suggested	Remarks
CPU	VT extensions	VT extensions, 64 bit, multicore	64-bit processors can run both 32 bit, and 64 bit instances; by default, Eucalyptus runs only 1 VM per CPU core on a Node
RAM	1 GB	4 GB	More memory means possibility for having larger guests, and more of them
Hard disk	5400 rpm IDE	7200 rpm SATA/SCSI	Since nodes are disk intensive, hard disks presents a performance bottleneck; consider faster versions (SATA, SCSI or SAN if the budget allows)
Disk space	40 GB	100 GB	Images are cached locally; please dimension this with care, since Eucalyptus is sensitive to running out of disk space
Network	100 Mbps	1 Gbps	Copying machine images (hundreds of MBs) is done over network; if the budget allows, consider 10 Gbps)

3.5 UEC Topologies

Finding an ideal configuration is a question of the needs and the budget available, and several UEC topologies are available for fitting the constraints[22].

In general, it is a question of finding the number of physical servers that would host the diverse UEC web service components (CLC/WS3/CC/SC/NC), and figuring an appropriate types of distributions of those components across the servers.
However, there will also be a need of another machine that can serve as a client machine. A client machine does not need to be a dedicated machine. Any OS that would include an internet browser could be installed on the required client machine. It would be used only for accessing the web interface of UEC, and for making remote control sessions for viewing and interacting with computer

[22] https://help.ubuntu.com/community/UEC/Topologies

desktops of instances, using i.e. Remote Desktop Protocol (RDP)[23] or VNC[24]. Those components – physical servers and client machine – are shown in Figure 7.

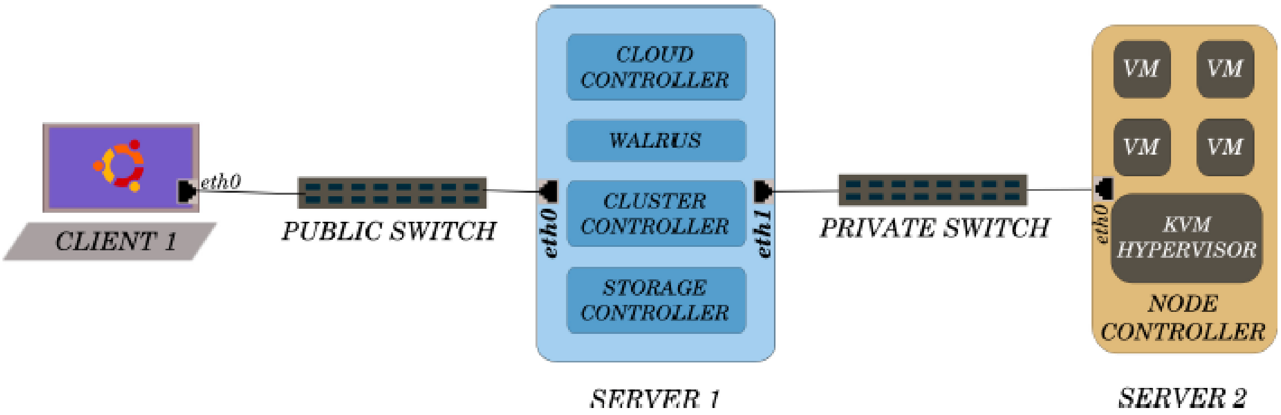


Figure 7 - Typical minimal Eucalyptus setup [19].

3.5.1 Discussion on different topologies

3.5.2 Implementation on a single server

In this configuration, all of the Eucalyptus components (CLC/WS3/CC/SC/NC) are implemented on the same machine.

Physical machine 1	CLC/WS3/CC/SC/NC

Implementing on one physical system is not a recommended topology for production use, but only for experimenting purposes. But implementing Eucalyptus on single machine is possible[25], though with some limitations.

Implementing inside a virtual machine would create an additional layer of virtualization, and KVM might not work properly, since VT enabled machines are needed.

A variation of implementing on one machine would be installing Ubuntu server on the physical machine, then installing VirtualBox, and under VirtualBox, run an Ubuntu virtual machine (VM) hosting CLC/WS3/CC/SC components. Since Ubuntu VM would not run virtual instances, there is no need for VT enabled machine. Then, install NC component on the physical host, and run the instances on it.

Physical machine 1	NC
Ubuntu VM	CLC/WS3/CC/SC

Furthermore, there is a network limitation implementing the whole solution on just one physical host. Since only supported network mode installing on one physical machine is either “system” or “static” networking modes, “managed” or “managed-novlan” networking modes cannot be used, thus disabling the use of some networking features, such as “security groups”, “elastic IPs” and “VM isolation”. To be able to use those features, Eucalyptus needs to be deployed on a minimum of two machines: one machine for front-end cloud components and one machine for the NC.

[23] http://en.wikipedia.org/wiki/Remote_Desktop_Protocol
[24] http://www.realvnc.com
[25] http://open.eucalyptus.com/wiki/FAQ#MinConfig

3.5.3 Implementation on 2 physical servers

This is the typical minimal Eucalyptus setup. All the user facing and back-end controlling components (CLC/WS3/CC/SC) are grouped on the first physical machine, and the NC component is installed on the second physical machine.

Physical machine 1	CLC/WS3/CC/SC
Physical machine 2	NC

This second physical machine is hosting virtual instances, and therefore should be appropriately dimensioned. This machine needs to be VT enabled also.
Scaling out possibility for this configuration is just adding more NCs.

3.5.4 Implementation on 3 physical servers

User facing components (CLC/WS3) are installed on the first physical server, and back-end controlling components (CC/SC) are on the second physical machine. The third physical machine is having the NC component installed.

Physical machine 1	CLC/WS3
Physical machine 2	CC/SC
Physical machine 3	NC

This third physical machine is hosting virtual instances, and thus should be appropriately dimensioned. This machine needs to be VT enabled also.
Scaling out possibility for this configuration is just adding more NCs.

3.5.5 Implementation on 4 physical servers

User facing components (CLC/WS3) are installed separately on the first (CLC) and second (WS3) physical server. Back-end controlling components (CC/SC) are installed on the third physical machine. The fourth physical machine is having the NC component installed.

Physical machine 1	CLC
Physical machine 2	WS3
Physical machine 3	CC/SC
Physical machine 4	NC

This fourth physical machine is hosting virtual instances, and should be appropriately dimensioned. This machine needs to be VT enabled also.
Scaling out possibility for this configuration is just adding more NCs.

3.5.6 Implementation on 5 physical servers

User facing components (CLC/WS3) are installed on the first physical server. Back-end controlling components (CC/SC) for the first cluster are installed on the second physical machine, and the third physical machine is having the NC component belonging to that cluster installed. Back-end controlling components (CC/SC) for the second cluster are installed on the fourth physical machine, and the fifth physical machine is having the NC component belonging to the second cluster installed.

Physical machine 1	CLC/WS3
Physical machine 2	CC1/SC1

Physical machine 3	NC1
Physical machine 4	CC2/SC2
Physical machine 5	NC2

These third and fifth physical machines are hosting virtual instances, and should be appropriately dimensioned. These machines needs to be VT enabled also.
Scaling out possibility for this configuration is adding more

- clusters, that is adding CC/SC and associated NC node(s), and
- NCs under clusters.

3.5.7 Other possibilities for implementation

Basically, there also other possible distributions (Figure 8) of web service components (CLC/WS3/CC/SC/NC) on available machines, depending on performance needs, or types of work distribution. If a component needs more processing power than storage capacity (like NC or CLC components), then one could group that components on a machine with greater processing power, or spread the components on several of such a machines. If accent is put on the storage (like in WS3 or SC components), then a machine with fast disk system with the capacity needed would be chosen.
But since UEC is very scalable, the choice of starting with a simple topology could be made, separating and distributing the roles as need for that arises.
One of scenarios would be starting with the implementation on two physical machines, separating user facing / back-end controlling components on one machine (CLC/WS3/CC/SC), and processing component (NC) on the other machine, as described in topology discussion above.
All depending on the application(s) that would be hosted on the cloud, a need for more processing power and/or storage capacity would appear occasionally. Then, if the need for more processing power would arise, another NC component could be installed on an additional machine, and added to the same cluster, controlled by the same CC. This process could be repeated every time our application reaches the processing power limitation. If the need for more storage capacity would arise, another SC component could be installed on an additional machine, and added to the same cluster, controlled by the same CC. This process could be repeated every time our application reaches the storage capacity limitation.

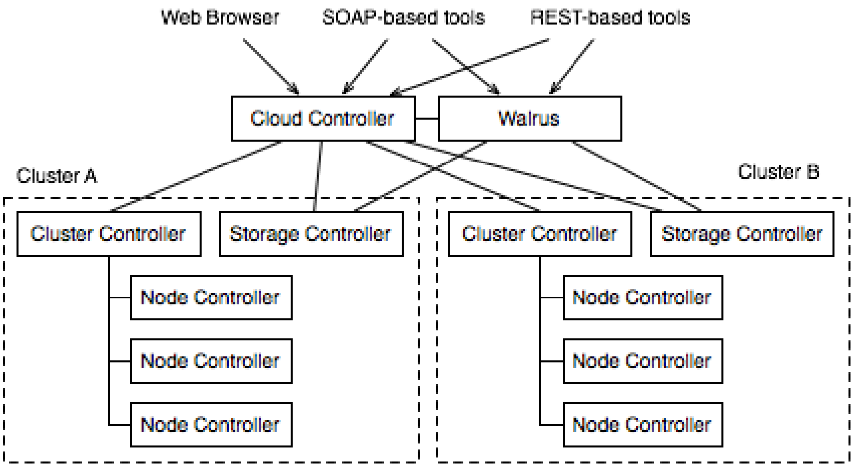


Figure 8 - Advanced Eucalyptus Setup[26]

[26] http://open.eucalyptus.com/wiki/EucalyptusAdvanced_v1.6

At the some point, adding SCs and NCs would cause that the CC could not cope with controlling all the SCs and NCs, and then another cluster could be added as mitigation, with new CC, SC and NC components being installed on additional machines. In that case, CLC/WS3/CC/SC components originally installed on one server would now be spread across three machines, one being responsible for the cloud (having CLC/WS3 components), and the other two for the two clusters (each having CC/SC components). The new CC and SC components would be used for controlling this new cluster and associated storage, and new NC would be supplying the processor power in this new cluster. Both the clusters would be a part in the same cloud, and controlled by the same CLC and WS3.

Then, if the need for more processing power in this new cluster would arise, another NC component could be installed on an additional machine, and added to this new cluster, controlled by the CC for that cluster. If more storage would be needed in this new cluster, additional SC would be added. Adding clusters to a private cloud in this way would, at some point, cause performance issues on the server controlling the whole cloud (having CLC and SW3 components installed). If this situation arises, a new machine can be added; and CLC and SW3 components could be hosted on two different machines in order to mitigating the performance related issues.

There are various reasons why a cloud should be divided into several clusters. One reason would be that different hypervisors would be running on the same cloud, and hypervisors are implemented per cluster. If there is a need of running both KVM and Xen, or VMware in the enterprise version, one cluster per hypervisor would be needed. Another reason can be the need of having single cloud management layer for spreading the cloud across an IT infrastructure based on two or more datacenters. An organizational structure and the needs implied by that structure could be yet another reason for having several clusters, e.g., having different clusters for different organizational units.

3.6 Network

Generally, a network solution can be viewed from two aspects: physical network infrastructures, and Eucalyptus network[27]. A physical network infrastructure, that designates the networking between components, can be designed in several ways: from the simplest, where all of the components are on the same, public network (Figure 9), to more complicated ones, like branching the communication between the nodes within the cloud on the separate network (Figure 10), and moving all storage communication (AoE, iSCSI) to a separate network.

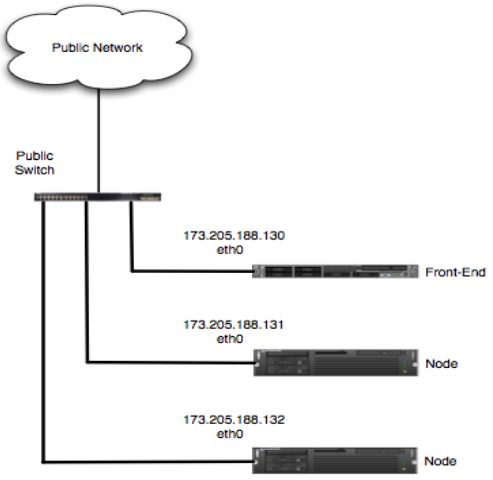


Figure 9 - Nodes connected directly to the public network

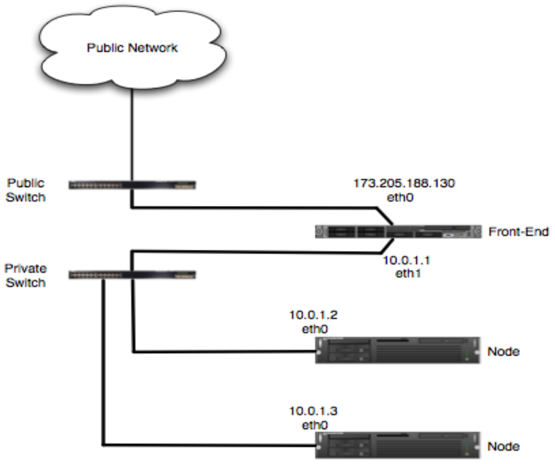


Figure 10 - Nodes on the private subnet, front-end on the public net

[27] http://open.eucalyptus.com/wiki/EucalyptusNetworkConfiguration_v2.0

A decision about the network is normally made by an IT department with the respect to different security and availability aspects. It is, though, strongly recommended not to run multiple clusters in the same broadcast domain. Each cluster should be in a separate domain, since the practical work during this project, experimenting with differently scaled architectures and different versions of Ubuntu, showed that a cluster belonging to one cloud controller would be visible by another cloud controller, with a possible interference as consequence, especially, if the same system accounts would be used. Designing Eucalyptus network, on the other hand, is a question of choice between the four different networking modes:

- Managed,
- Managed-NoVLAN,
- System, and
- Static.

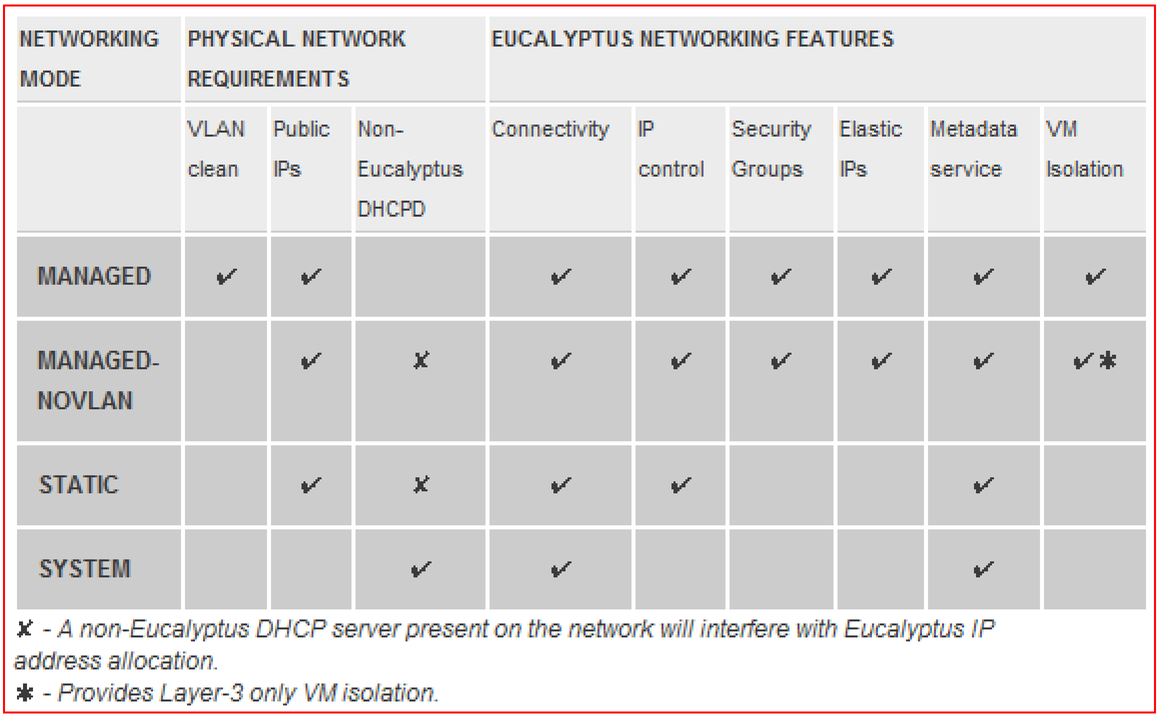
They accommodate Eucalyptus deployment on a variety of physical network infrastructures. One thing to bear in mind is that if running Eucalyptus in multi-cluster mode, it is strongly recommend that identical networking mode is configured for all clusters.

A managed mode[28] is one having the most features: it provides security groups, VM isolation, and elastic IPs and Metadata service. Managed-NoVLAN provides the same features as Managed mode, except VM isolation between instances. System mode has no virtual subnets—VM instances are directly bridged with the NC machine's physical Ethernet device, appearing on the physical network. This mode is least demanding for the physical network, and is dependent on the DHCP server installed on it. A static mode gives more control over instances' IP addresses, where each instance gets its own MAC/IP pair at start, where IP is originating from a Eucalyptus' DHCP.

Eucalyptus networking features include connectivity, IP control, security groups, elastic IPs, metadata service and VM isolations. All modes provide IP connectivity, and in all modes except system, Eucalyptus assigns IP addresses to VMs. Security groups are set of networking rules (in fact, firewall rules, like i.e., allowing incoming SSH connections on TCP port 22) that apply to all of the instances that are member of the group. Elastic IPs is a pool of dynamically assigned IP addresses that are assigned to instances, and this feature is available in Managed and Managed-NoVLAN modes. An AWS-compatible metadata service is provided by Eucalyptus, which allows access to instance-specific information from inside a running instance (information like instance's public and private IP address, reservation ID, user-data and a launch index). Isolation of network traffic between different security groups is available in a managed mode (Managed-NoVLAN mode guarantees isolation only with security groups on different subnets – "layer-3 isolation"). This is enforced using VLAN tag per security group, thus, effectively preventing eavesdropping between security groups - though, network traffic between instances inside a security group is open.

Before choosing a networking mode for Eucalyptus network configuration, it is important to understand the interrelationship between the availability of features and the requirements for the physical network (Figure 11), since some features can only be implemented if the physical network meets certain requirements.

[28] Underlying physical network must be VLAN clean, since Eucalyptus provides and manages its own VLAN tagging



NETWORKING MODE	PHYSICAL NETWORK REQUIREMENTS			EUCALYPTUS NETWORKING FEATURES					
	VLAN clean	Public IPs	Non-Eucalyptus DHCPD	Connectivity	IP control	Security Groups	Elastic IPs	Metadata service	VM Isolation
MANAGED	✓	✓		✓	✓	✓	✓	✓	✓
MANAGED-NOVLAN		✓	✘	✓	✓	✓	✓	✓	✓✱
STATIC		✓	✘	✓	✓			✓	
SYSTEM			✓	✓				✓	

✘ - A non-Eucalyptus DHCP server present on the network will interfere with Eucalyptus IP address allocation.

✱ - Provides Layer-3 only VM isolation.

Figure 11 - Relationship between Eucalyptus networking modes and corresponding requirements for the underlying physical network[29]

3.7 Implementation possibilities / hardware vendors

Hardware implementation is strongly dependent on the budget. During this project, several main hardware producers (HP, IBM and DELL) were contacted with a proposal of establishing a start-up solution as a fully functional private cloud infrastructure with the possibility of further extension. The proposed extension was envisaged by scaling out as scaling up is not an option in modern hardware systems, where servers are built from integrated modules that cannot be easily separated moreover, the servers are also a part of a bigger unit that is mounted onto a rack. In that way, a system is easy to maintain, since if hardware error would occur, one would just change the whole module – in this case the whole server, putting a new server in the unit.

HP's initiative called "Fast Track Private Clouds"[30], based on their "Converged Infrastructure" with "HP CloudSystem Matrix"[31] and "Data Storage" seemed quite interesting, but further exploration revealed that it was too big hardware solution for starting a private cloud on a smaller scale. With IBM, the situation was somewhat similar – they have "IBM System x Private Cloud Offering"[32], again too big solution for starting a private cloud on a smaller scale. At the same time, it became clear that only DELL had a solution that would fit the purpose [22], being suitable for small scale startup. It seemed that DELL starter kit would be a starting point based on the following information:

- DELL has starter-kit for UEC, with short implementation time, and good scalability;
- DELL is provider for Amazon and SalesForce.com, so the technology is already tested and running on servers;
- The servers have been customized to run Ubuntu-based cloud services, with hardware compliant with Ubuntu

[29] http://open.eucalyptus.com/wiki/EucalyptusNetworkConfiguration_v2.0
[30] http://www.hp.com/hpinfo/newsroom/press/2010/100830a.html
[31] http://h18004.www1.hp.com/products/blades/components/matrix/index.html
[32] http://www.redbooks.ibm.com/redpapers/pdfs/redp4731.pdf

- The system could be used for implementing private clouds, but also as a test and pre-prod environment for organizations developing applications to run on Amazon Web Services (Eucalyptus' API is compatible with AWS's API); since the system has a preconfigured testing and development environment, it could be used to test the applications locally before uploading them to Amazon's paid service.

Starter-kit for Dell UEC SE Cloud Solution comes standard with the following hardware configuration [23] (Figure 12):

- The Cloud Frontend Server – a PowerEdge™ C2100 running all shared UEC-related service (Walrus, CLC, CC and SC);
- The Infrastructure Server - a PowerEdge C2100 running all cloud deployment and storage functions
- Compute Nodes - a PowerEdge C6100 system that embeds four discrete compute nodes in a single enclosure;
- Network Switch - a Dell PowerConnect™ 6248 network switch optimized for iSCSI;
- Optional: KVM 1U drawer (mouse, keyboard, display);
- Optional: server rack, for organizations that don't already have rack.

Dell UEC SE Cloud Solution is a single-cluster (availability zone), four compute-node IaaS cloud. Each (embedded) compute node has two CPUs, and each CPU has four cores, which makes a total of eight cores per compute node. Having four of such compute nodes embedded in one hardware unit, the total number of cores for such unit is 32 cores. Scaling out the cluster would simply require adding such unit to an existing cluster without the need of a lot of rack space and cabling.

This implementation comes with an iSCSI storage server for storage – infrastructure server. If facing shortage on disk capacity, scaling out would be adding one of such infrastructure servers. This cloud in its original, starter kit, setup is dimensioned to run up to 64 virtual instances at one time (that depends naturally on an instance's size). The hardware can be placed in a standard 19 inch rack. Without the optional KVM taking 1 rack unit, the hardware implementation of this solution takes 7 rack units of rack space.

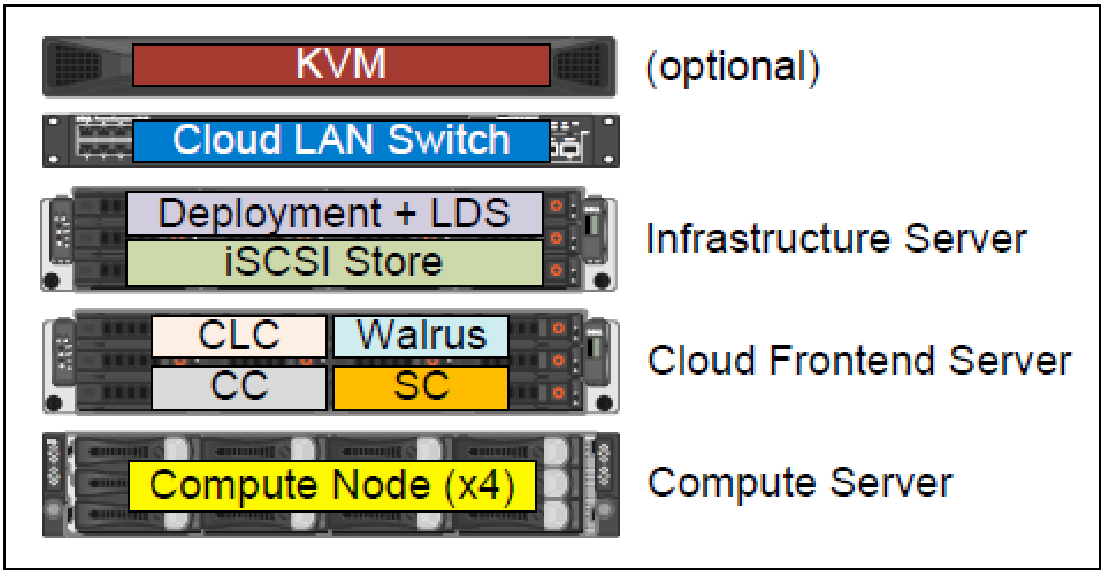


Figure 12 - Cloud solution – rack servers [23]

Compactness of this setup is attractive for the organizations starting with private cloud implementation, especially because one would not have much overhead concerning the hardware setup and maintenance because of its modularity.

3.8 Installation in virtual environment

For preliminary test purposes, and as a kind of “Proof of Concept” (PoC), an installation [9] [24] [19] of UEC in a virtual environment is performed. This environment was used for testing the scalability options. The environment topology of servers and network is shown in Figure13.

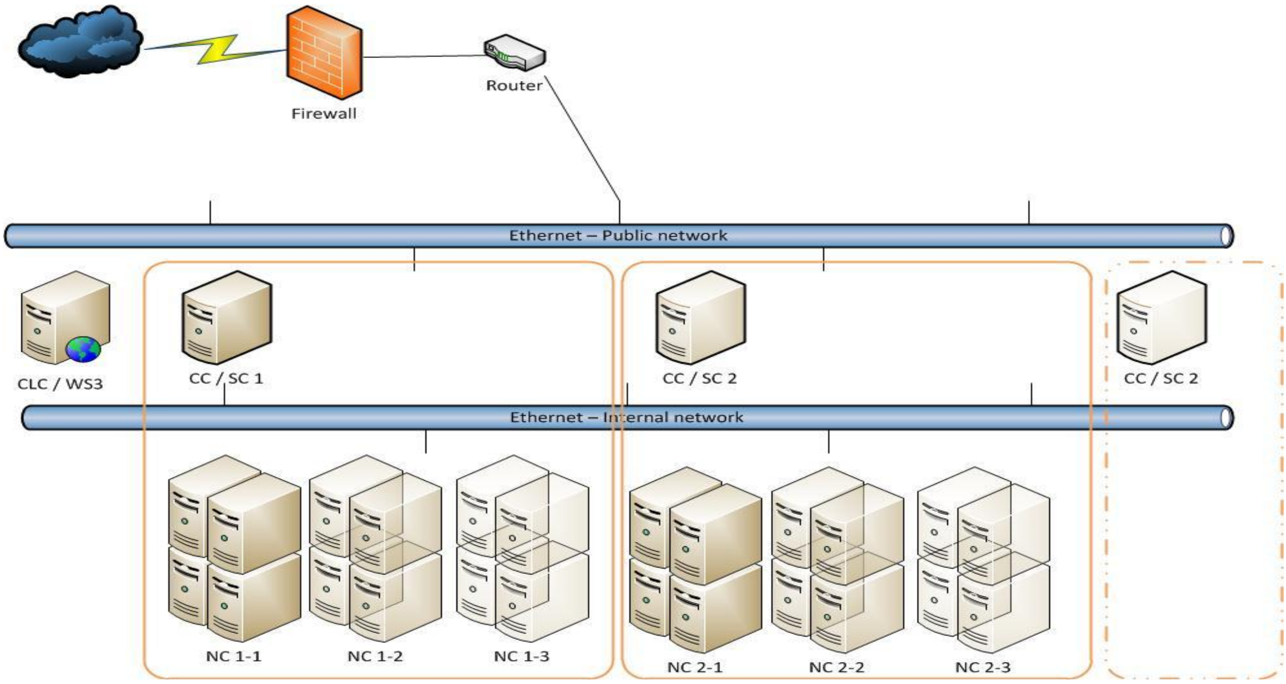


Figure 13 – Virtual environment topology

The installation in virtual environment was used as a way to test different scale-out possibilities, since provisioning of the virtual machines with associated networking is quick and easy. A reinstallation of the whole environment could also be done quickly. The support for snapshots served as a possibility for reverting if things would go wrong. As first, scaling out of a cluster was tested, adding node controllers to a cluster. Afterwards, scaling out of the whole cloud was tested, adding a new cluster (CC and SC), and associated node controllers. For full details, refer to this report [25].

3.9 Installation on physical environment

For preliminary test purposes, and as a kind of “Proof of Concept” (PoC), an installation of UEC on physical servers, using physical networking was performed. Figure 14 shows the servers and network topology used for this installation.

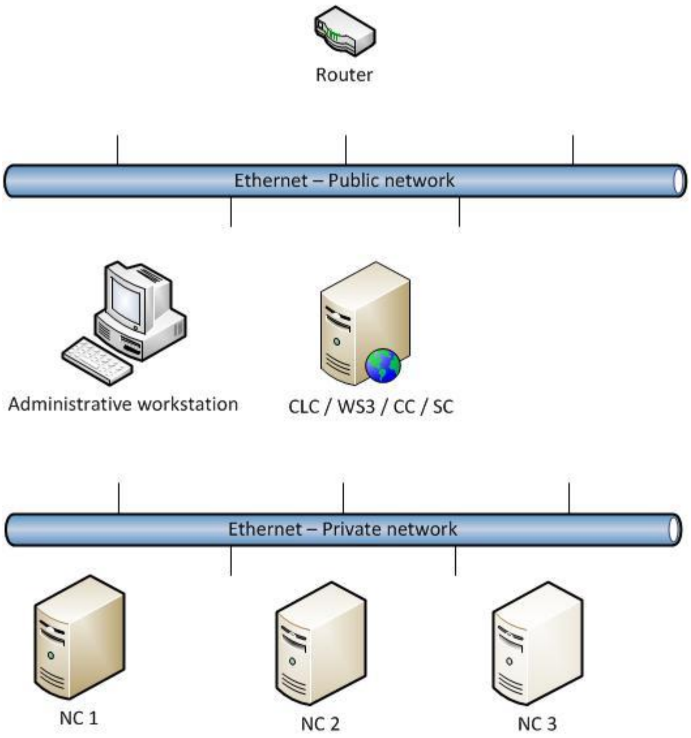

Router
Ethernet – Public network
Administrative workstation
CLC / WS3 / CC / SC
Ethernet – Private network
NC 1
NC 2
NC 3

Figure 14 – Physical environment topology

The installation in a physical environment was used for testing the installation and establishing a private cloud. For reasons of simplicity of an initial private cloud implementation, the cloud and cluster roles were put on one server, and the NC role was installed on the other server. It was envisaged that there would be an eventual scaling out caused by performance problems would be done either by adding node controllers, or adding a whole new cluster (the latter option would also feasible if another department would have a cluster of its own). After running several machines, a NC’s limit would be reached, and then there will be a need to scale out adding some more NC’s. When storage and cluster performance would be reached, there would a need to scale out adding a whole cluster. Both of these scale-out possibilities have been described in the companion report [25].

Besides processing power, the storage limits would also be reached soon, and SAN (iSCSI) based storage should be added to all servers. The SAN is suggested implementation on iSCSI technology, since it is cheaper and less complex to implement and maintain[33] than the Fiber Channel technology, and the performance for the two technologies are comparable, especially for a smaller organizations[34], and with Ethernet switches nowadays reaching the speed of 10 Gbps. There are also open source solutions available that match the proprietary solutions and having almost the same

[33] http://www.webopedia.com/DidYouKnow/Computer_Science/2006/iSCSI_Fibre_Channel.asp

[34] http://invurted.com/iscsi-vs-fibre-channel-storage-performance/

features, for example, an open source solution that would match “NetApp”[35] is a Nexenta-based NAS, SAN and Backup Server solution called “Napp-it”[36].

To make the whole environment independent of the network infrastructure and the environment where it would be “plugged in”, architectural decision was made to install a firewall / router in front of the whole private cloud, and then to open the ports needed as the need occur. The advantage of this approach is security, and the fact that the environment can be easily moved network-wise, but the drawback is that there is an administrative burden giving the access to the instances and the whole environment from the “outside world”, and the firewall being a bottleneck. The firewall used is open source FreeBSD based pfSense[37]. We also installed an additional machine as an administrative workstation in the environment. The machine was based on Ubuntu Linux using Mozilla Firefox browser as a Windows based environment appeared to be giving some unexpected problems while controlling the private cloud and its instances.

During the design of the whole system, an implementing redundant architecture was considered, but after some research, we decided not to follow that path. At this stage, there is no redundancy available in UEC by design. If a server with an UEC role would get a software or hardware error which would render it incapable of functioning as a part of the cloud, a new server with the same role would be the only solution, and all the data that are present on that server would be lost, and would have to be restored. That is why the recommendation of using the SAN as a storage solution was made, mainly for the persistent storage (storing items like files, databases, etc.). Though, in an academic environment, especially in scientific computing where the main load are batches, loosing instances (which are stateless) would not be a big damage – if the data was stored on the persistent storage, there is a need of bringing some new instances, and run the batch job(s) again. Hence, not having redundancy, we are not able to have any failover possibilities. If a server would be faulty, the only solution would be installing a new server to replace the failed one. For full details, refer to this report [26]. We have also produced a report [27] describing the troubleshooting steps during installation of Ubuntu based private cloud.

[35] http://www.netapp.com/us/products/
[36] http://www.napp-it.org/index_en.html
[37] http://www.pfsense.org

4. Summary and recommendations

4.1 Cloud computing options available

An organization that is considering implementing the cloud technologies can have several choices for using cloud computing technologies:

- Licensed / Enterprise private cloud solutions,
- Specially tailored solutions,
- Private cloud based on open source,
- Public Cloud.

Nowadays, most organizations have an existing IT infrastructure usually based on licensed versions of private cloud solutions (like it is a case with Corporate IT at University of Copenhagen that uses VMware). The advantages are a number of available features and the support options. Quota management is also present, mostly as "soft quota", where a flag is raised if a user reaches predefined usage threshold, rather than closing that user's access to different services. The disadvantages include relatively high level of complexity and relatively more expensive.

Organizations that need to provide their users with special and tailored solutions, or need very large or non-standard software installations could go for self-developed implementation. That requires time and effort for implementation, but at the end there is a solution that can greatly fit the needs. This solution's vulnerabilities are the level of expertise needed for designing, deploying and maintain the solution, and keeping it up to date with trends and advancing both in organization's are of interest, and with systems and software used for implementation.

Opting for an open source based solution, like using Open Eucalyptus and Open Nebula, is also an available option. Those solutions would provide computing power and storage, but not more than that. A whole layer of middleware is (for the time being) missing – services like database services, load balancers and queuing services should be implemented by organization itself. The current solutions are also relying heavily on system administrator's expertise, and need a high level of control of the network topology. There is no support for quota management.

For organizations just starting with cloud computing, it may be more appropriate to implement solutions using a public cloud provider. The public cloud services from well known cloud provider are pretty stable and are rich with features. Moreover, the level of accessibility of the public cloud providers enables average system administrators to support the organizational needs without having to acquire a lot of new expertise. It could be a natural choice, especially at the start, when organizations want to start with their projects as soon as possible. With the public clouds offers, there is no need to have specialist knowledge as there are communities built around a cloud service provider that means both users and administrators could seek help and knowledge if required.

4.2 Choosing a solution

Choosing a solution is dependent mainly on the usage pattern. An analysis should be performed to find out if the load is evenly distributed over a period of time, or if there are temporary peaks of load. If user jobs are of a predictable character, with flat computing, network and storage usage, natural choice would be implementing a private cloud. If peaks are occurring, a hybrid solution should be implemented, where a private cloud is used to cover the flat part, and public cloud resources are used during peak times. In our case, that was one of the reasons for choosing UEC based on Eucalyptus. Having the same API as Amazon, choosing UEC would not limit us only to private cloud – if after the initial implementation a need for “bursts” would occur, our solution and implemented application would be ready for run on the Amazon’s infrastructure during the peak times (assuming that there are no problems with security and legal considerations).

4.3 Private cloud – advantages and disadvantages

Setting up a private cloud has a steep learning curve for the whole organization, for both users and administrators. Although highly scalable, with possibility for installing at a number of sites, it seems that the work needed to install and maintain large stage deployments is too big, and significant manpower would be needed, both in terms of time and skills. There is a need of more than average level of skills required for installing, managing, and using a private cloud. Long-running larger private clouds based on Eucalyptus seems to be fragile [18]. Moreover, private cloud needs current hardware. To mitigate all the work for manual firing up the instances, or services installed on those instances, an independent scripting language could be implemented for the purpose of automation.

Comparing public vs. private clouds - private clouds are more customizable, but have greater installation cost, and lower flexibility. Public clouds are more mature, being well documented, rich with features, and easy to use. Private clouds are still at an early stage, and not much predictable. They are still too hard to install, manage, and maintain for a regular administrator. However, the effort required for running a private cloud is having a downward tendency. It is expected that the level of effort required for implementing solutions on a private cloud and the public cloud will be similar at some point in the near future. For organizations not familiar with the technology used in private cloud implementations and short of funds and skills, it would be advisable to go for a public cloud implementation at the beginning while learning and working with the provide cloud technology with the aim of building the required knowledge and skills.

Although implementing a private cloud may not be an option for an organization at the moment, just neglecting the private cloud solutions and their viability would at some point result in different projects throughout organization finding alternative ways to cloud, avoiding organizational politic. That would definitely result in lower cost efficiency and less structured usage of the IT resources. Some of the key takeaways from this report are:

- Implement a private cloud at affordable smaller scale, on a current hardware;
- Keep the software platform, hardware and knowledge updated;
- Monitor and analyze costs, available features and complexity compared to the budget, needs and internal resources available;
- Asses the private cloud implementation possibilities based on results of those analyses.

5. Acknowledgment

The work reported in this report has been partially funded by the Danish Agency for Science, Technology and Innovation under the project "Next Generation Technology for Global Software Development", #10-092313.

6. References

[1] M. Armbrust*, et al.*, "A View of Cloud Computing," *Communication of ACM,* vol. 53, pp. 50-58, 2010.

[2] A. Lenk*, et al.*, "What is Inside the Cloud? An Architectural Map of the Cloud Landscape," presented at the Workshop on Software Engineering Challenges of Cloud Computing, Collocated with ICSE 2009 Vancouver, Canada, 2009.

[3] L. Vaquero*, et al.*, "A break in the clouds: towards a cloud definition," *SIGCOMM Computer Communications Review,* vol. 39, pp. 50-55, 2009.

[4] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, USA2009.

[5] R. Grossman, "The Case for Cloud Computing," *IEEE Computer,* vol. 11, pp. 23-27, 2009.

[6] Q. Zhang*, et al.*, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications,* vol. 1, pp. 7-18, 2010.

[7] A. Trolle-Schultz, "Cloud Computing Course F2011," *IT University of Copenhagen,* 2011.

[8] R. Belso and F. Orellana, "Interview at Danish Centre for Scientific Computing (DCSC)," *Danish Centre for Scientific Computing (DCSC),* 2011.

[9] S. Wardley*, et al.*, "Ubuntu Enterprise Cloud Architecture," Canonical, USA, Technical ReportAugust 2009.

[10] "SAM-DATA/HK Magasinet," 2011.

[11] F. Baiardi and D. Sgandurra, "Secuing a Community Cloud," presented at the IEEE 30th International Conference on Distributed Computing Systems Workshops, 2010.

[12] M. Ali Babar and A. Chauhan, "Toward a Framework for Migrating Software Systems to Cloud Computing," IT University of Copenhagen, Denmark TR-2011-139, 2011.

[13] D. H. Strowd and G. Lewis, "T-Check in System-of-Systems Technologies: Cloud Computing," Softwar Engineering Institute, Carnegie Mellon University, Technical Report CMU/SEI-2010-TN-009, September 2009.

[14] Khajeh-Hosseini A*, et al.*, "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS," presented at the IEEE 3rd International Conference on Cloud Computing, Miami, USA, 2010.

[15] "Microsoft TectEd, cloud related sessions," *Atlanta, USA,* 2011.

[16] "SharePoint Saturday Copenhagen," *IT University of Copenhagen,* 2011.

[17] CANONICAL, "Ubuntu Case Study - KnowledgeTree Moves to the Cloud," 2011.

[18] k. Edlund*, et al.*, "Practical cloud evaluation from a nordic eScience user perspective," presented at the Proceedings of the 5th international workshop on Virtualization technologies in distributed computing, San Jose, California, USA, 2011.

[19] J. D*, et al.*, "Eucalyptus Beginner's Guide - UEC Edition," *CSS,* 2010.

[20] L. Bass*, et al.*, "Quality Attribute Design Primitives and the Attribute Driven Design Method," in *Proceedings of the 4th International Workshop on Product Family Engineering*, Bilbao, Spain, 2001.

[21] "Ubuntu - An introduction to cloud computing," 2011.

[22] J. Jackson, "Dell Releases Ubuntu-powered Cloud Servers," *PCWorld,* 2011.

[23] "Deployment Guide - Ubuntu Enterprise Cloud on Dell Servers SE," *DELL,* 2011.

[24] "Building a Private Cloud with Ubuntu Server 10.04 Enterprise Cloud (Eucalyptus)," *OSCON,* 2010.

[25] Z. Pantić and M. Ali Babar, "Installing and Scaling out Ubuntu Enterprise Cloud in Virtual Environment," IT University of Copenhagen, Denmark TR-2012-154, 2012.

[26] Z. Pantić and M. Ali Babar, "Installing Ubuntu Enterprise Cloud in a Physical Environment," IT University of Copenhagen, Denmark TR-2012-155, 2012.

[27] Z. Pantić and M. Ali Babar, "Troubleshooting during Installing Ubuntu Enterprise Cloud," IT University of Copenhagen, Denmark TR-2012-156, 2012.