Hybrid Clouds: Comparing Cloud Toolkits

Seminar Paper

Chair in Information Systems III
Prof. Dr. Martin Schader
University of Mannheim

April 2010

by
Philipp C. Heckel
heckel@informatik.uni-mannheim.de

Declaration of Authorship

I, Philipp C. Heckel, declare that this seminar paper and the work presented in it are my own.

Furthermore, I declare that this seminar paper does not incorporate any material previously submitted for a degree or a diploma in any university; and that to the best of my knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text.

Signed: ______________________________

Date: ______________________________

Contents

Abbreviations **iv**

1 Introduction **1**

1.1 Scope . . . 2

1.2 Structure . . . 2

2 Cloud Computing **3**

2.1 Status Quo . . . 3

2.2 Definitions and Key Characteristics . . . 4

2.3 Classifications . . . 4

2.3.1 Service Models: Abstraction Classes . . . 5

2.3.2 Deployment Models: Exposure Classes . . . 5

3 Hybrid Clouds **6**

3.1 Opportunities . . . 6

3.2 Challenges and Issues . . . 7

3.2.1 Cost . . . 7

3.2.2 Security and Data Confidentiality . . . 8

3.2.3 Availability . . . 9

3.2.4 Interoperability . . . 10

4 Cloud Toolkits **12**

4.1 Market Overview . . . 12

4.2 Technical Requirements and Restrictions . . . 15

4.2.1 Hardware Requirements . . . 16

4.2.2 Operating Systems and Software Restrictions . . . 16

4.3 OpenNebula and Eucalyptus . . . 17

4.3.1 Eucalyptus . . . 18

4.3.2 OpenNebula . . . 18

5 Conclusion **20**

Bibliography **ix**

Abbreviations

AMI .... Amazon Machine Image
API .... Application Programming Interface
AWS .... Amazon Web Services
CEO .... Chief Executive Officer
CPU .... Central Processing Unit
CRM .... Customer Relationship Management
DDoS .... Distributed Denial of Service
e.g. .... exempli gratia – *for example*
EBS .... Elastic Block Storage – *Amazon product*
EC2 .... Elastic Computing Cloud – *Amazon product*
EMI .... Eucalyptus Machine Image
GUI .... Graphical User Interface
HA .... High Availability
i.e. .... id est – *that is*
I/O .... Input/Output
IaaS .... Infrastructure as a Service
IT .... Information Technology
KVM .... Kernel-Based Virtual Machine
LVM .... Logical Volume Management
NFS .... Network File System
OS .... Operating System
PaaS .... Platform as a Service
RHEL .... Red Hat Enterprise Linux
RHEV .... Red Hat Enterprise Virtualization
S3 .... Simple Storage Service – *Amazon product*
SaaS .... Software as a Service
SAN .... Storage Area Network
SOAP .... Simple Object Access Protocol
SSH .... Secure Shell
VLAN .... Virtual Local Area Network
VM .... Virtual Machine
VPN .... Virtual Private Network

1 Introduction

In the last few years, the importance of the Internet has risen constantly and made it indispensable for businesses and most individuals to be on-line around the clock. One of the greatest drivers of this development was and still is the shift of the traditional one-to-many Web to an advanced, participatory version of the Word Wide Web. Rather than only making editorial information accessible to many users, the Web 2.0 encourages participation and enables user generated contributions. Leveraging this new paradigm, services like Flickr, Facebook, or Twitter have become very prominent examples for this development [3, 45].

An essential part of this evolution, but mostly hidden to the end-consumer, is the set of tools that enable these large scale applications. *Cloud computing* is a relatively new technology that serves as underlying architecture for most of these platforms. By providing virtualized computing resources as a service in a pay-as-you-go manner, cloud computing enables new business models and cost effective resource usage. Instead of having to maintain their own data center, companies can concentrate on their core business and purchase resources when needed. Especially when combining a privately maintained virtual infrastructure with publicly accessible clouds in a *hybrid cloud*, the technology can open up new opportunities for businesses and help consolidating resources. However, since cloud computing is a very new term, there are as many definitions of its components as there are opinions about its usefulness. Most of the corresponding technologies are only a few years old and the toolkits lack of maturity and interoperability [20].

This seminar paper introduces the basic concepts of cloud computing and discusses the technical requirements for setting up a hybrid cloud. It briefly looks into security concerns and outlines the status quo of current cloud technologies. In particular, it evaluates several existing cloud toolkits regarding its requirements, occurring problems and interoperability.

1.1 Scope

Cloud computing is a very up-to-date research topic and is being heavily discussed by the IT research community as well as by IT managers from the business perspective [45].

Business side analysis often focus on using the on-demand payment models of public clouds to cut costs, and to minimize investments of the in-house IT infrastructure. By comparing the operating costs of a data center with the expenses necessary for buying cloud resources, many studies discuss the trade-off between in-house IT provisioning and IT outsourcing from a cost perspective [4]. This mostly includes detailed analysis of the key business factors such as required computing time, data center utilization, or peak usage, but also covers more risk-oriented topics like security or service level agreements.

Even though the business perspective is highly relevant for a complete analysis of the topic, the scope of this seminar paper mostly comprises technical aspects. This particularly includes the definition of relevant terms as well as the classification of different cloud approaches. Furthermore, it analyzes different cloud toolkits and describes how to set up and deploy a cloud in a privately owned data center.

1.2 Structure

The seminar paper is divided in five chapters.

In order to present a broad view of cloud computing, chapter 2 discusses the status quo of the current cloud market and briefly skips through the most important steps in its young history. It then identifies key characteristics of the technologies and introduces relevant terms.

Chapter 3 mainly focuses on hybrid cloud computing by identifying existing opportunities, challenges and issues of the technology. This particularly includes topics like cost, security, availability, and interoperability.

After giving a detailed view of hybrid cloud computing and its issues, chapter 4 focuses on actual possibilities to deploy a cloud. Therefore, it first introduces existing toolkits and gives a short market overview of commercial and non-commercial software. It then looks into hardware and software requirements and finally presents the two open source toolkits OpenNebula and Eucalyptus in greater detail.

Chapter 5 closes with a short discussion and a brief summary of the seminar paper.

2 Cloud Computing

This chapter shortly introduces the status quo of cloud computing and sets it in context to other computing paradigms. It furthermore gives an overview over the basic concepts and defines the relevant terms.

2.1 Status Quo

As newest concept in the development of distributed computing, cloud computing is often believed to be *"the next step in the evolution of the Internet"* [14]. As foundation and enabler for *Software as a Service*, it delivers computing resources over the Internet and provides elastic scalability for any kind of application. While cluster and grid computing already allowed multiple computers to work together on complex tasks in a distributed manner, the cloud concept extends this idea even further: instead of regarding individual machines, cloud computing treats resources as a *utility* [29]. That is computing time and storage are provisioned on-demand and paid per usage without the need for any upfront commitment [4].

As one of the first commercial providers of cloud services, Amazon launched a beta version of its *Elastic Computing Cloud (EC2)* in August 2006 and announced production stability in October 2008 [1, 5]. Google followed with a public beta of *App Engine* in April 2008 [23, 34], and Microsoft made its cloud platform *Windows Azure* publicly available in February 2010 [38, 40]. The well known alternatives to the commercial solutions are several open source cloud toolkits. Prominent examples include OpenNebula [44], a project started by researchers of the University of Chicago and Madrid in 2008, as well as the Eucalyptus cloud software, initiated by the University of California, Santa Barbara in 2007 [43].

Even though there are already many commercial and open source cloud solutions, all of them are fairly young and have yet to prove their acceptance and durability. According to the Open Cloud Manifesto, the technology *"is still in its early stages, with much to learn and more experimentation to come"* [14]. This particularly includes challenges that yet need to be overcome, e.g. data security within the cloud, or interoperability between different clouds.

2.2 Definitions and Key Characteristics

Due to the actuality of the topic, there are several opinions about what cloud computing and its corresponding terms comprises. Some experts see the technology as *"one of the foundations of next generation computing"* [45, Tim O'Reilly, CEO of O'Reilly Media], others believe that the term is just a buzz word to define *"everything that we currently do"* [28, Larry Ellison, CEO of Oracle Corp.].

However, while the term is being criticized, there are still many intersecting definitions describing the technology. Some are broader than others and include not only the technical part, but also the services enabled by the cloud, i.e. SaaS applications. For the Laboratory of Distributed Systems of the University of California, Berkeley, for instance, *"cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services."* [4]. Not contrary to that definition, but limiting the term to hardware and software, IBM describes cloud computing as *"an emerging computing paradigm where data and services reside in massively scalable data centers and can be ubiquitously accessed from any connected devices over the Internet."* [33].

In this seminar paper, cloud computing is defined by the three key characteristics shared by many experts [4, 36, 33, 14, 29]:

- **Resource abstraction**: resources inside the cloud are not directly observable by the cloud user, but are virtualized using technologies like Xen or KVM, and can be accessed via an application programmable interface (API).
- **Elastic capacity**: the cloud appears to users as a pool of infinite capacity, being able to allocate and free resources on-demand. Scaling up and down avoids over- and under-utilization and thereby allows an optimal load.
- **Utility-based pricing**: storage, CPU time and network bandwidth are charged by the hour using a pay-per-use pricing model. Resources can be allocated almost instantaneously without any upfront commitment.

2.3 Classifications

Similar to the attempts to define the term cloud computing, the categorization of it is rather difficult, if not *"impossible in the currently rapid evolution of the cloud landscape"* [30]. However, many papers classify cloud systems by their level of abstraction and their exposure to the Internet.

2.3.1 Service Models: Abstraction Classes

In order create the illusion of infinite resources and elasticity, virtualization technology is needed. Depending on how abstracted resources are, different *service models* are differentiated [36, 4, 29]:

- **Software as a Service (SaaS)**: at the highest level of abstraction, users are mostly unaware of the fact that are using cloud-enabled applications, and are hence not able to control the underlying resources. Instead, they simply use client interfaces such as web browsers. A popular example is the salesforce.com CRM system [53].
- **Platform as a Service (PaaS)**: users are able to develop and deploy applications within the provider's hosting environment, e.g. a Java application framework. Low-level resources are not controlled by the cloud user. Prominent example is the Google App Engine [23].
- **Infrastructure as a Service (IaaS)**: at the lowest level of abstraction, cloud users have access to virtualized resources such as processing time, networking or storage. They are provided virtual machines and can run arbitrary software. Famous example is Amazon EC2 [1].

2.3.2 Deployment Models: Exposure Classes

Not every cloud is available for public use: depending on the level of exposure to the Internet, the following *deployment models* are differentiated [36, 44, 29]:

- **Public Cloud**: the cloud infrastructure is publicly accessible via an API. It is hosted by a cloud provider who sells its capacity using a pay-per-use payment model. All of the above mentioned examples are public clouds.
- **Private Cloud**: the cloud infrastructure is hosted within the data center of an organization and used by local users only. It focuses on providing a flexible virtualized infrastructure rather than on selling the cloud resources.
- **Hybrid Cloud**: the hybrid cloud approach extends the private cloud model by using both local and remote resources. It is usually used to handle flash crowds by scaling out when the local capacity is exhausted. This so called cloudbursting enables highly elastic environments [44]. The key difference between private and hybrid clouds is *"the extension of service provider-oriented low cost cloud storage to the enterprise"* [31]. That is remote cloud resources are seamlessly integrated in the private cloud, and thereby create a hybrid cloud.

3 Hybrid Clouds

Public clouds have become increasingly popular in the last few years since they allow almost instant resource provisioning and fast scaling without having to maintain a data center. As one of the first public cloud solutions, Amazon's EC2 has strongly contributed to this development. However, not only the cloud community is growing rapidly, but also the number of critiques is increasing. Especially in terms of data security and privacy, but also in other topics (such as availability, vendor lock-in, cost, or interoperability), public clouds carry inherent risks.

An obvious yet not trivial solution to these issues is the use of both private and public delivery models, and combining them to a *hybrid cloud.* This chapter briefly analyzes the existing opportunities and obstacles of cloud computing – particularly regarding hybrid clouds.

3.1 Opportunities

In a hybrid cloud, a company maintains its own private cloud, i.e. a virtualized data center, and can scale out to a public cloud if needed. Moving from a traditional data center to a hybrid cloud approach brings many benefits to businesses [4]:

- **Optimal utilization**: in typical data centers, only 5% to 20% of the available server resources are actually used [4, pg. 10]. Because peak loads are up to ten times higher than the average load, servers are mostly idle – generating unnecessary costs. Hybrid clouds can increase server utilization by scaling out to public resources to handle flash crowds.

- **Data center consolidation**: instead of having to provide capacity for worst-case scenarios, a private cloud only requires resources for the average case. The option to burst out allows server consolidation and hence the reduction of operating costs. In particular, this includes the costs for hardware, power, cooling, maintenance, and administration.

- **Risk transfer**: while the companies themselves are responsible for keeping their data center and private cloud up and running, the public cloud provider has to ensure a high uptime for its service. Using a hybrid cloud model, *"the risk of mis-estimating workload is shifted from the service operator to the cloud vendor"* [4, pg. 11]. Most cloud providers have service level agreements ensuring an uptime of more than 99.9% per year, i.e., a downtime of max. 9 hours per year [2, 39].

- **Availability**: ensuring high availability in the corporate data center is difficult and expensive, because it requires redundancy, backups, and geographic dissemination. Especially in companies where IT is not the core business, the expertise in this area is rather limited. In a hybrid cloud environment, the public cloud can scale up or take over operations completely if the company's data center is unavailable due to failures or DDoS attacks.

3.2 Challenges and Issues

Even though hybrid clouds offer a great value proposition and enable many opportunities, the number of challenges and issues is also very high. Especially due to its still evolving nature, cloud computing has many unsolved economic and technical issues. The following sections discuss the most important issues briefly.[1]

3.2.1 Cost

One of the most obvious obstacles, and certainly the most important one from the business perspective, is the fact that hybrid cloud infrastructures require both a local data center *and* additional remote resources from a cloud provider. That is, the often mentioned benefit of cloud computing – the independence of a data center – does not hold true for hybrid environments. In fact, hybrid cloud infrastructures have to factor in the setup and operating cost for a data center (e.g. hardware, power, cooling, maintenance) as well as the usage-based costs of the cloud provider. Depending on utilization, data center cost and the costs of the cloud provider, businesses have to decide whether or not moving to the cloud is profitable.

In its technical report *Above the Clouds*, the UC Berkeley proposes a simple model to compare the expected profits of in-house computing with the profits of using public cloud resources [4]. Even though their model is based on very strong assumptions, it identifies the important key characteristics that influence the decision:

- **Pay separately per resource**: most applications do not use available resources equally, but rather use one of them extensively. Some applications are CPU inten-

[1] This seminar paper only discusses a subset of known obstacles of cloud computing. Extensive lists can be found in [4, 20, 32].

sive, others might be storage or bandwidth-oriented. Depending on the resource-type, external providers might offer better conditions than a local data center can offer.

- **Power, cooling and physical plant costs**: depending on how expensive the private data center is, local applications have to factor in the costs for power, cooling and other plant expenses.
- **Operations costs**: cloud environments have lower hardware operations costs, because data centers are virtualized and the risks of outages can be moved to external providers. The operations costs of software management, however, stays the same in IaaS environments and decreases the with an increase in the abstraction level (lower costs in SaaS environments).
- **Utilization**: profits and costs strongly correlate with the degree of data center utilization. While external cloud providers include operations costs in the usage costs, the local data center costs must be set in relation to the utilization.

Depending on the usable capacity of the local data center and the usage costs of the cloud provider, businesses have to decide how many public resources to use. Even though the presented characteristics can help indicating the amount of capacity to buy from external providers, infrastructure decisions are not only profit driven, but have to consider other factors.

3.2.2 Security and Data Confidentiality

One of these factors is the security of data and information. According to studies conducted by Colt in 2009 [32, pg. 47], over 60% of the IT decision makers are still insecure regarding cloud security. Robert Biddle, Professor at the Carleton University in Ottawa, anecdotally depicts security in the cloud like this [20]:

> *"Leslie Lamport described a distributed computing system with the following warning: 'You know you have one when the crash of a computer you've never heard of stops you from getting any work done.' Nowadays there is an additional risk: a machine you've never heard of pretends to be another machine you've never heard of, and steals all your data."*

Even though his description is provocatively worded, the message is true for most cloud environments. In fact the potentially dangerous scenarios of cloud computing are numerous.

While servers in a privately owned data center are physically under the control of the IT department, virtual machines inside a cloud are located anywhere in the world and controlled by the cloud provider. Leaving business critical machines to an external instance not only requires solid service level agreements, but also trust in the provider's capabilities and fidelity. But the physical location of the servers is not the only issue: more problematic is that the server is not part of the company's network anymore, but of the public Internet. In classical corporate networks, the different IT components such as logon servers, directory services or file servers are hidden in the inner perimeter and shielded with a firewall. Inside a cloud, these mechanisms to secure the machines do not apply, and other security measures have to be installed.

In fact potential attackers do not need to break through the firewall anymore, but can easily access the same network or even the same physical machines by simply renting virtual machines from the cloud provider. While the topology and software of an in-house data center is unknown to outside instances, the cloud reveals its technologies to potential attackers by definition. That is attackers can analyze the server environment, network traffic and possible hypervisor bugs without having to break through the inner perimeter. Even though the VMs on a physical host and their virtual networks (VLANs) are isolated from each other, a bug in the hypervisor can void these security measures. As only barrier between the guest and the host system, the hypervisor is a single point of attack and its correctness is crucial [32].

In hybrid cloud environments companies can avoid these issues at least to a certain extent: the most obvious solution is to not move any sensitive information to the cloud and hence use public resources only for non-critical calculations and services. Other measures to increase security are for example installing a virtual private network (VPN), or disk encryption. However, even with all the suggested measures, the data is still located in a foreign environment, and a company can never be sure about what happens to them.

3.2.3 Availability

Cloud providers are specialized on providing a scalable and fault-tolerant environment with a good quality of service. To ensure a high uptime, they introduce high-availability systems and maintain several data centers all over the world.

In most cases not having to cope with an HA system is beneficial to businesses because it saves hardware and maintenance costs (cmp. section 3.1). But when the public cloud system is not operating as expected, companies are at the provider's mercy. System failures and complete outages are not only an issue with small cloud providers, but also hit the global players such as Amazon or Google. Amazon AWS for instance was unreachable several times in its young history, e.g. 48 hours in October 2007, two hours in February

2008, and eight hours in July 2008 [29]. The last big outage in December 2009 was caused by a power failure in the data center in North Virgina and lasted for *"several hours"* [26]. Another outage in January 2010 only hit a few high-capacity instances and was caused by a routing device in the same data center [7].

In cloud supported environments, availability is an even more important issue than it is in traditional data centers. Nils Magnus, author at the Linux Magazin, believes that the total availability decreases with an increase of distributed components. If a system requires all local and remote services to be available to operate properly, using public cloud resources can certainly lower the total uptime [32]. However, if different public cloud providers are used as an alternative, high-availability can still be maintained. In the opinion of the UC Berkeley, *"the only plausible solution to very high-availability is using multiple cloud computing providers"* [4].

In regard to hybrid clouds, availability strongly depends on how public resources are integrated in the complete system. In case the public cloud is only used for cloudbursting, i.e. the local resources are extended in peak times, the use of multiple providers can limit the risk significantly. However, if the cloud resources are interweaved with important business processes, the impact of a public cloud failure is considerably higher. An example of the latter case is the Ruby on Rails platform provider Heroku [24], which just recently had to deal with a crash of approximately 44,000 hosted applications. As a startup company, Heroku is completely dependent of the availability of Amazon's Elastic Computing Cloud. Due a problem in EC2, the 22 rented virtual machines vanished and had to be redeployed after Amazon fixed the problem [7].

3.2.4 Interoperability

Another often discussed issue [4, 20, 14] of current clouds is the fact that the different cloud systems do not work well together. In fact the young age of the concept has led to various incompatible systems that only slowly approach each other in terms of interoperability. Beginning from the hypervisor level up to the application programming interfaces, currently available clouds differ fundamentally.

At the lowest level, clouds consist of interconnected virtualized hosts using hypervisors like Xen, KVM or VMware's ESX. The hypervisor is responsible for managing the physical hardware and mapping them to several virtual machines. The technologies are developed and maintained by different organizations, and are only compatible to a certain extent. That is a virtual machine that has been created and deployed for one hypervisor does not necessarily run on a different one. Even though the interoperability on the hypervisor level has increased in the last years, there are still different virtual disk and virtual machine file formats. However, the collaborative work of XenSource/Citrix and VMware has pushed the development of common file formats towards standards (e.g. the Open Virtualization Format [15]).

While many hypervisors already support several VM formats, current infrastructure providers do not use this potential interoperability on the next higher level. An example for this compatibility gap is the market's strongest player Amazon: instead of using its market strength to establish standard APIs and virtual machine formats, Amazon holds on to its EC2 API and Amazon Machine Image (AMI) format. Other competitors such as GoGrid [22] or ElasticHosts [18] also use their own APIs and virtual machine formats [21, 17], so that switching from one cloud hosting provider to another is currently only possible if virtual machines are converted and API calls are adapted.

To close this gap, the Open Cloud Manifesto [14] defines six basic principles that aim towards establishing and adapting standards where ever possible. While over 300 companies already signed the document (including the several cloud providers such as VMware, Rackspace, or GoGrid), many big names are missing on the list of supporters: Amazon, Google, Salesforce and Microsoft refused to sign the manifesto. All of them have proprietary cloud software or APIs, and are competitors on the cloud market. They intensively advertise their cloud solutions and try to establish their software as de-facto standards (lock-in). Larry Dignan, Editor in Chief of ZDNet, believes that the providers are in an *"API war"* and that *"it is far too early [for them] to sign off on a manifesto when the cloud is still in its infancy"* [27].

The issue of interoperability is particularly important in hybrid cloud environments because they integrate different cloud solutions. While a completely compatible system would allow exchanging cloud providers and VM images transparently, current hybrid cloud toolkits have to deal with numerous existing incompatibilities. Instead of using a standardized API and file format, toolkits usually implement a public interface and translate them to public cloud API calls. VM image formats are currently completely incompatible and cannot be handled by toolkits like Eucalyptus or OpenNebula. That makes migration between private and public cloud impossible, and hence restricts the flexibility of hybrid clouds significantly. Brian J. Dooley, cloud expert and author at Information Management Online, even believes that the *"vision of the hybrid cloud is, at present, a projection. Currently, interoperability is somewhat limited at various points, including at the virtualization hypervisor level; data transfer also remains problematic, as is integration between applications in separate clouds."* [16].

4 Cloud Toolkits

After discussing the opportunities and obstacles of cloud computing, this chapter introduces existing toolkits for deploying private and hybrid clouds, i.e. software to manage a virtual private infrastructure. The first part gives a brief overview of commercial and non-commercial toolkits, and outlines their differences as well as similarities. The second part of the chapter analyzes the technical requirements for deploying a hybrid cloud, and then presents the two open source toolkits OpenNebula and Eucalyptus in greater detail.

4.1 Market Overview

Many companies use the hype around cloud computing to advertise their products as cloud-enabling software. And in fact, the vague definition of the term allows a broad interpretation of what this comprises, and makes it rather difficult to determine relevant products. Table 4.1[1] and the following paragraphs briefly introduce the most important virtual infrastructure management tools, i.e. cloud toolkits. The scope of this analysis will only include IaaS solutions that allow deploying private and/or hybrid clouds. It does not attempt to be complete, but simply represents a snapshot of well-known toolkits.

VMware, the biggest player in the virtualization market, offers several cloud-enabling pieces of software: its flagship *vSphere*, formerly known as *VMware Infrastructure*, is a full data center virtualization solution. It is based on the ESX hypervisor that manages a single host. By combining many ESX hosts and connecting them to with an internal network, the virtualized servers form a private cloud [46]. Even though VMware advertises vSphere as a hybrid cloud solution, its hybrid abilities are very limited: vSphere is able to scale out only if the public cloud provider also uses vSphere, i.e. using Amazon's EC2 for cloudbursting is not supported [35, 54]. Because of this major deficit, vSphere rather classifies as pure data center virtualization software rather than a hybrid cloud toolkit. However, in terms of its features, vSphere outperforms its open competitors by orders of magnitude. In particular, VMware offers enterprise features such as high availability, fault-tolerance, or distributed resource scheduling [46].

With its data center virtualization solution *Red Hat Enterprise Virtualization (RHEV)* [52], Red Hat also focuses on enterprise customers. The RHEV hypervisor is based on the

[1] The presented table is mainly based on [54], [6], [52], [59], [9], [47], [42], [56], [51], [11], [48], and [57].

	VMware vSphere	RHEV	Citrix XenServer	Hyper-V R2	Euca-lyptus	Nimbus	Open Nebula	oVirt
Hypervisor	VMware	KVM	Xen, *Hyper-V*	Hyper-V, Xen	Xen, KVM, *VMware*	Xen	Xen, KVM, VMware	Libvirt / KVM
VLAN	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Image Mgmt.	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Scheduling	Yes	Yes	N/A	N/A	Limited	External	External	No
Live Migr.	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High Avail.	Yes	Yes	*Yes*	Yes	No	No	No	No
Hybrid Cloud	No	No	No	No	Partially	Partially	Yes	No
Admin GUI	Yes	Yes	Yes	Yes	No	No	No	Yes
Requires Intel VT / AMD-V	No	Yes	only for proprietary guests	No	if KVM hypervisor is used	only for proprietary guests	if KVM hypervisor is used	if KVM hypervisor is used
Officially supported Guest OSs	Windows, Linux, Solaris, NetWare	Windows, RHEL	Windows, RHEL, CentOS, SLES, Debian	Windows, SLES, RHEL	Depends on Hypervisor	Depends on Hypervisor	Depends on Hypervisor	Depends on Hypervisor
API	vCloud API	N/A	XenServer XML RPC	Hyper-V WMI	EC2 subset	EC2 subset & own WS	EC2 subset & OCCI	No
License	Proprietary	Proprietary	Proprietary	Proprietary	BSD *Proprietary*	Apache 2	Apache 2	GPLv2
Annual Cost	up to \$4400 per CPU	up to \$750 per socket	Free / *up to \$5500 per host*	up to \$3000 per CPU	Free / *N/A*	Free	Free	Free

Table 4.1: Comparison of Cloud Toolkits. In case there is a free *and* premium version, terms in *italic font* indicate features that are only available in the premium version.

Kernel-Based Virtual Machine (KVM) and is built from a subset of Red Hat Enterprise Linux (RHEL). RHEV's features include live migration, virtual networking, and image management [49]. Similar to VMware vSphere, it additionally provides a sophisticated Administration User Interface and enhanced enterprise features.

As relatively new player in the cloud market, Microsoft entered the competition with its virtualization solution *Hyper-V*. Hyper-V provides similar functionalities as its competitors, but is not yet as advanced in terms of fault-tolerance and other virtualization-specific features. Instead it rather focuses on traditional server features and goes deeper with alert management or monitoring. In the new R2 revision, it added live migration and a cluster-aware file system. Hyper-V is available as standalone hypervisor, or included in the Windows Server 2008 R2 operating system [55].

Another big player in the virtualization sector is Citrix Systems: with its *XenServer*, it competes with vSphere and RHEV in the enterprise segment and therefore mainly focuses on private cloud provisioning for data centers. It is based on the Xen hypervisor and allows the deployment of a virtual data center. Like its competitors, it also features live migration (XenMotion), and a simple GUI-based multi-server management. Since February 2009, XenServer is available for free [12], including most of its enterprise features. The commercial extensions *Citrix Essentials* allow using Microsoft's hypervisor Hyper-V, and enable additional features such as high availability and e-mail alerting.

Compared to the commercial products, the open source cloud toolkits *Eucalyptus* [19], *Nimbus* [42], *OpenNebula* [44] and *oVirt* [47] have a very limited feature set. None of them provides high availability or fault-tolerance, and only oVirt includes a usable graphical user interface. The projects are in a very early development state and have not reached production stability yet. Most projects lack of a complete up-to-date documentation and have a complicated installation process. Deploying a private or hybrid cloud using these toolkits encompasses a lot of Linux scripting and requires significant knowledge about other software and technologies (e.g. about external schedulers, lease managers, or virtual file formats). In fact some experts believe that the open source solutions are *"light years away"* from their commercial competitors [32].

The most prominent example of the open source toolkits is *Eucalyptus*, a project initiated by the UC Berkeley in 2007. Eucalyptus is an acronym for *"Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems"* and is now run by Eucalyptus Systems, an open source company founded to promote and further develop the software. Eucalyptus features an open software package as well as a commercial product. While the open product supports the Xen and KVM hypervisors, Eucalyptus EE additionally works with VMware's ESX/ESXi, i.e with vSphere. Eucalyptus implements several Amazon AWS interfaces (including EC2, S3 and EBS) and thereby allows the creation of a EC2-compatible private cloud. Like all presented solutions, it also features live migration, VLAN support, as well as image management.

Similar to Eucalyptus, the *Nimbus* project also focuses on providing a private IaaS cloud with AWS-compatible interfaces. It was founded by researchers at the University of Chicago in 2008, and mainly concentrates on scientific computing. The initial implementation was based on the Xen hypervisor, but newer versions also support KVM. It is very adjustable in terms of third party software and can be configured to use external schedulers such as the Sun Grid Engine (SGE). In future releases, the developers plan to include an EC2 back-end so that EC2 instances can be integrated transparently [41].

Opposed to Eucalyptus and Nimbus, *OpenNebula* explicitly advertises itself as a hybrid cloud toolkit. The project was started in a cooperation of the Universities of Chicago and Madrid in 2008, and focuses on integrating different clouds in a single system. It uses a plugin-based technology to support a variety of different hypervisors and public cloud providers. It currently works with Xen, KVM or VMware and can integrate Amazon EC2 as well as ElasticHosts as cloud provider. Compared to the other projects, OpenNebula corresponds most to the hybrid cloud definition.

oVirt is part of Red Hat's Emerging Technology project and was first released in 2008. The project describes itself as *"virtualization management framework"* for realizing a virtual data center. It is based on the *libvirt* library and can therefore potentially support a variety of hypervisors. However, the current stable version of the software only works with KVM. Compared to the other open source projects, the oVirt's strongest part is its Web-based administration interface: while OpenNebula, Eucalyptus and Nimbus only provide command line interfaces and APIs, oVirt allows managing multiple hosts and resource pools with a simple Web GUI.

Currently the commercial cloud software providers such as VMware, Red Hat and Citrix have a more advanced feature set and provide fully-integrated enterprise solutions. With enhanced GUI tools and seamless integration of multiple servers, they all allow a relatively easy management of virtualized data centers. In contrast, the open source solutions are far behind in terms of completeness or stability. However, while they do not feature a complete enterprise-ready software package, they are more advanced when it comes to hybrid cloud computing: Eucalyptus, OpenNebula[2] and Nimbus, for instance, allow the creation of a hybrid cloud at least to some extent. While Eucalyptus and Nimbus reach this goal by simply imitating the EC2 interface, OpenNebula integrates remote resources transparently. Nimbus and Eucalyptus can hence be controlled with the same tools as Amazon's Elastic Computing Cloud.

4.2 Technical Requirements and Restrictions

The currently available cloud software allows the creation of a very flexible virtualized computing infrastructure and can bring great benefits in a modern IT environment (cmp. chapter 3.1). However, as indicated in chapter 3.2.4, clouds have much room for improvement in terms of interoperability. Most cloud software is designed to work in a well-defined infrastructure and only if all requirements are met, the cloud works as expected. Especially the commercial cloud software (such as VMware vSphere or RHEV) strictly defines the supported hardware and software, but also open source solutions make high demands on the systems.

[2]Details on OpenNebula and Eucalyptus will be discussed in chapter 4.3.

4.2.1 Hardware Requirements

The hardware requirements of cloud solutions are very different in most cases, and depend not only on the used hypervisor, but also on the cloud management software. In order to guarantee service levels to their software packages, VMware, Citrix and Red Hat define very large hardware compatibility lists of supported processors, storage and I/O systems [13, 58, 50]. For companies with incompatible hardware, switching to a virtualized infrastructure can hence become very expensive because new hardware might be necessary. The non-commercial projects do not explicitly define supported hardware, but instead simply specify minimal hardware requirements such as memory or CPU speed.

These requirements often include CPU virtualization technologies such as Intel VT/VT-x or AMD-V. Depending on the hypervisor and the type of virtualization, the host system's processors must provide these virtualization extensions to function. While KVM only supports hardware-assisted virtualization, i.e. with CPU virtualization functionalities, ESX- and Xen-based cloud toolkits also allow full virtualization and paravirtualization [11, 57]. In case of Xen, however, paravirtualization is only possible for Linux guests. For proprietary operating systems, only hardware-assisted virtualization is possible. Table 4.1 indicates exact compatibilities.

Besides CPU compatibility, especially the commercial cloud solutions rely on a very specific hardware configuration and topology. Some products require certain network layouts, or other components to be present. Citrix XenServer, for instance, requires VM images to reside on a SAN storage *"to use advanced platform features such as resource pools, shared storage, live migration and high availability"* [10]. VMware's vSphere only supports live migration for a limited set of processors: administrators have to *"make sure that the source and destination hosts have compatible processors"* [58].

While the big vendors have very high requirements, they can at least guarantee that the system works as expected if the listed hardware is used. For the open source projects, however, compatibility between hosts requires a trial-and-error approach.

4.2.2 Operating Systems and Software Restrictions

In addition to the numerous hardware requirements, most cloud toolkits restrict the number of usable operating systems and other software significantly. While the open source solutions do not specifically list the officially supported operating systems, the vendors of commercial virtualization software only certify a very small number of guest OSs.

Especially Microsoft and Red Hat mainly focus on supporting their own operating systems and hence only provide a very limited choice: Hyper-V supports several Windows versions, including Windows Server 2000 – 2008 R2 and Windows XP – 7 (excluding the Home editions) [37]. However, in terms of other OSs such as Solaris or Linux, it only supports two enterprise Linux distributions for a single virtual CPU (Red Hat Enterprise Linux 5 and SUSE Enterprise Linux 10–11). Red Hat's RHEV officially supports even fewer operating systems: besides its own RHEL, it is only compatible with certified versions of Windows XP, 2003 and 2008. The other commercial solutions have similar compatibility lists: and even though vSphere and XenServer officially support a broader range of OSs, they limit the choices significantly.

Compared to the commercial vendors, the open source solutions seem to have a much larger OS support. KVM-based cloud toolkits, for instance, have been reported to work with over 100 guest operating systems of seven OS families, including Windows, Debian/Ubuntu, Red Hat/Fedora, BSD, and Solaris [25]. However, the two paradigms pursue completely different goals: while the commercial software aims towards stability and productive operations, its open competitors are rather focused on supporting a large number of operating systems.

In addition to the limited amount of certified guests, all of the available VI toolkits enforce the use of specific APIs and command-line tools to manage the cloud. These tools are mostly product-bound and require a certain amount of expertise. Since they cannot be used to control other clouds, switching to a different software can become very expensive.

Similar to the required hardware, the OS and software restrictions of current cloud toolkits are very noticeable and have to be considered before deploying a private or hybrid cloud. When businesses have to choose a cloud solution, they not only have to consider obvious issues like availability, security and cost, but also face many problems when it comes to compatibility of hardware and software: current cloud toolkits are still far away from being natively supported by any machine. Instead, they are only usable in certain configurations and topologies, using certified hardware and a limited set of software.

4.3 OpenNebula and Eucalyptus

As two of the most promising open source solutions for deploying private and/or hybrid clouds, OpenNebula and Eucalyptus have reached great publicity in the last year. Even though they both enable the conversion of a regular data center in a virtualized infrastructure, they follow completely different approaches.

4.3.1 Eucalyptus

Eucalyptus is an open source software framework that implements an IaaS environment. It can deploy private or public clouds, and *"gives users the ability to run and control entire virtual machine instances deployed across a variety of physical resources"* [43]. Since October 2009, Eucalyptus is part of the Linux distribution Ubuntu Server, rebranded as *Ubuntu Enterprise Cloud*.

The Eucalyptus API is compatible to Amazon EC2 and hence makes it possible to control both Amazon and Eucalyptus instances with the same tools. Its main objectives are to provide a platform for testing applications before they are moved to Amazon's infrastructure, as well as to manage and control large collections of distributed resources [43].

In general, Eucalyptus (and the included storage daemon *Walrus*) emulate EC2 and S3 by providing the same SOAP and Query interfaces as Amazon, and by acting similar to the real Amazon cloud. However, even though the external APIs are mostly identical, the interior of the Eucalyptus cloud is rather different: Amazon's EC2 is based on a modified version of the Xen hypervisor and uses its own image format (Amazon Machine Image, AMI). In contrast, Eucalyptus can run on Xen, KVM or VMware ESX and also ships with a different VM format (Eucalyptus Machine Image, EMI). With the exception of the APIs, both systems are hence completely incompatible.

Eucalyptus does not advertise itself as hybrid toolkit, but rather as private cloud software. And in fact it strongly depends on the definition of hybrid cloud computing whether or not it qualifies as such: Eucalyptus does not integrate remote resources transparently in the private infrastructure, and it provides no tools to extend the local capacity via external cloud providers. Instead it simply emulates the EC2 infrastructure, and can thereby serve as foundation for hybrid solutions. It is not designed to be a hybrid cloud software, but rather *"to bridge between public and private clouds to enable hybrid cloud infrastructures"* [16].

4.3.2 OpenNebula

Opposed to Eucalyptus, and currently the only software on the market that describes itself as hybrid cloud toolkit is OpenNebula. And in fact, compared to the other solutions it corresponds most to the definition of a hybrid cloud. While other toolkits either introduce their own proprietary infrastructure (vSphere, RHEV, XenServer, Hyper-V) or emulate others (Eucalyptus, Nimbus), OpenNebula transparently integrates external resources in the cloud.

It has a flexible component-based structure and includes several predefined drivers for information management (monitoring), image and storage management (e.g. via NFS, LVM, or SSH), as well as to support several hypervisors (KVM, Xen, VMware). In addition to the numerous private cloud drivers, OpenNebula also integrates drivers for Amazon EC2 and ElasticHosts, and can be easily extended to support other cloud providers. With its support for EC2, OpenNebula can be configured to use Amazon's infrastructure for cloudbursting, or use other EC2 compatible systems (such as Eucalyptus or Nimbus).

An OpenNebula cloud typically consists of a front-end node for administration purposes (i.e. for managing hosts and images), as well as of several cluster nodes to execute the VM images. The hosts are controlled by the front-end either via command-line tools, or via well-defined programming interfaces.

Even though the current development status of OpenNebula is far away from a production-ready product, it *"differentiates itself from the other platforms in the sense that it has been designed to federate existing technologies"* [8]. By leveraging the advantages of other virtualization software, it combines them to a strong hybrid cloud tool with high potential.

5 Conclusion

Cloud computing has reached great publicity in the last few years and has become a new trend in the IT industry. And even though its popularity increases faster than ever, it has to overcome a lot of issues and faces many challenges. Especially in hybrid cloud environments, incompatibilities between the available cloud solutions prevent a broad adoption throughout all businesses.

This seminar paper introduced the basic concepts of cloud computing and discussed the opportunities and issues of current cloud solutions. By comparing several existing toolkits, it classified commercial and non-commercial software regarding their possible application scenarios and presented technical requirements and restrictions. By taking a close look at the two open source solutions OpenNebula and Eucalyptus, it set the vision of hybrid clouds in context to the current status quo of the technology.

The results of this seminar paper show that hybrid cloud computing is a very young and poorly researched area, and that both commercial and non-commercial products have a lot of catching up to do. Current virtual infrastructure and cloud software are not built for hybrid environments and hardly feature any cloudbursting support at all. While the amount of private cloud toolkits and public cloud providers steadily increases with the popularity of the topic, only very few products are able to leverage their full potential by combining the two paradigms in a hybrid cloud.

Especially the big commercial solutions such as vSphere or Hyper-V currently largely focus on providing private cloud solutions to enterprise customers rather than integrating hybrid cloud support. None of them features the option to scale-out to public cloud providers such as Amazon or ElasticHosts, or interfaces with other external services. Even though the open source solutions are forced to interact more with external providers, their integration of public clouds is rather limited and has yet to mature technically. The reason for this is among other things the lack of standards for virtual machine formats and communication protocols. This makes interaction between toolkits difficult and hinders the vendors to enhance interoperability and reduce incompatibilities.

Cloud computing is a cutting edge technology and evolves rapidly day by day. Both private and public clouds have proven to work perfectly in enterprise environments. However, the power of the concept lies in a combination of the two. The only missing elements to unveil the numerous advantages are to introduce standards and address security and trust issues. Only if cloud vendors are able to solve these issues, cloud users can leverage the full power of hybrid clouds.

Bibliography

[1] Amazon, Inc. Amazon Elastic Computing Cloud. http://aws.amazon.com/ec2/.

[2] Amazon, Inc. Amazon EC2 Service Level Agreement. http://aws.amazon.com/ec2-sla/, October 2008.

[3] Paul Anderson. What is Web 2.0? Ideas, technologies and implications for education. Technical report.

[4] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[5] Jeff Barr. Big Day for Amazon EC2. http://aws.typepad.com/aws/2008/10/big-day-for-ec2.html, October 2008.

[6] Constantino Vázquez Blanco. Execution of SGE Clusters on top of Hybrid Clouds using OpenNebula. http://www.opennebula.org/_media/constantino_vazquez_-_opennebula_-_executing_sge_clusters_on_top_of_hybrid_clouds_using_opennebula.ppt, September 2009.

[7] Carl Brooks, Techtarget. Heroku learns the hard way from Amazon EC2 outage. http://searchcloudcomputing.techtarget.com/news/article/0,289142,sid201_gci1378426,00.html, January 2010.

[8] Damien Cerbelaud, Shishir Garg, and Jeremy Huylebroeck. Opening the clouds: qualitative overview of the state-of-the-art open source VM-based cloud management platforms. In *Middleware '09: Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, pages 1–8, New York, NY, USA, 2009. Springer-Verlag New York, Inc.

[9] Citrix Systems, Inc. XenServer. http://www.xenserver5.com/.

[10] Citrix Systems, Inc. XenServer Getting Started. http://www.xenserver5.com/videos/Getting_Started_with_XenServer/Getting_Started_with_XenServer.htm.

[11] Citrix Systems, Inc. XenServer Requirement and Specifications. http://www.citrix.com/English/ps2/products/subfeature.asp?contentID=1681139.

[12] Citrix Systems, Inc. New Citrix XenServer Release Makes Enterprise-Class, Cloud-Proven Virtualization Free for Everyone. http://www.citrix.com/English/NE/news/news.asp?newsID=1687130, February 2009.

[13] Citrix Systems, Inc. XenServer Hardware Compatibility List. http://hcl.xensource.com/, April 2010.

[14] Cohen et al. Open Cloud Manifesto, Spring 2009.

[15] Distributed Management Task Force, Inc. (DMTF). Open Virtualization Format Specification. http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf, February 2009.

[16] Brian Dooley. Architectural Requirements Of The Hybrid Cloud. http://www.information-management.com/news/hybrid-cloud-architectural-requirements-10017152-1.html, February 2010.

[17] ElasticHosts. API Specification. http://www.elastichosts.com/cloud-hosting/api, April 2010.

[18] ElasticHosts. ElasticHosts. http://www.elastichosts.com/, April 2010.

[19] Eucalyptus Systems, Inc. Eucalyptus. http://open.eucalyptus.com/.

[20] Steven Fraser, Robert Biddle, Scott Jordan, Kate Keahey, Bob Marcus, E. Michael Maximilien, and Dave A. Thomas. Cloud computing beyond objects: seeding the cloud. In Shail Arora and Gary T. Leavens, editors, *OOPSLA Companion*, pages 847–850. ACM, 2009.

[21] GoGrid. API Specification. http://wiki.gogrid.com/wiki/index.php/API, April 2010.

[22] GoGrid. Cloud Hosting. http://www.gogrid.com/cloud-hosting/, April 2010.

[23] Google, Inc. Google App Engine. https://appengine.google.com/.

[24] Heroku, Inc. Heroku Official Website. http://heroku.com/, April 2010.

[25] KVM. Guest OS Support Status. http://www.linux-kvm.org/page/Guest_Support_Status, 2010.

[26] Lance Whitney, CNET News. Amazon EC2 cloud service hit by botnet, outage. http://news.cnet.com/8301-1009_3-10413951-83.html, December 2009.

[27] Larry Dignan, ZDNet. Can you have an Open Cloud Manifesto without Amazon, Google, Salesforce and Microsoft? http://blogs.zdnet.com/BTL/?p=15331, March 2009.

[28] Larry Ellison, CEO of the Oracle Corp. What The Hell Is Cloud Computing? http://youtube.com/?v=0FacYAI6DY0, September 2008.

[29] Kevin Lee. Cloud Computing Lecture, December 2009.

[30] Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai, and Thomas Sandholm. What's inside the Cloud? An architectural map of the Cloud landscape. In *CLOUD '09: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pages 23–31, Washington, DC, USA, 2009. IEEE Computer Society.

[31] Steve Lesem. Cloud Storage for the Enterprise - The Hybrid Cloud. http://cloudstoragestrategy.com/2010/01/cloud-storage-for-the-enterprise---part-2-the-hybrid-cloud.html, January 2010.

[32] Nils Magnus. Wetterwechsel: Daten, Netze und Sicherheit in die Cloud auslagern. *Linux Magazin*, 05/10:46–48, May 2010.

[33] Mark O'Neill, CTO, Vordel. Connecting to the cloud: Leverage the cloud in applications. IBM developerWorks, http://www.ibm.com/developerworks/xml/library/x-cloudpt1/, March 2009.

[34] Paul McDonald. Introducing Google App Engine. http://googleappengine.blogspot.com/2008/04/introducing-google-app-engine-our-new.html.

[35] Laurianne McLaughlin. Does VMware vSphere solve IT's worries about the cloud? http://www.infoworld.com/d/virtualization/does-vmware-vsphere-solve-its-worries-about-cloud-326, April 2009.

[36] Peter Mell and Tim Grance. The NIST Definition of Cloud Computing, Version 15. http://csrc.nist.gov/groups/SNS/cloud-computing/, October 2009.

[37] Microsoft Corporation. Virtualization with Hyper-V: Supported Guest Operating Systems. http://www.microsoft.com/windowsserver2008/en/us/hyperv-supported-guest-os.aspx.

[38] Microsoft Corporation. Windows Azure. http://www.microsoft.com/windowsazure/.

[39] Microsoft Corporation. Windows Azure Compute SLA, Version 1.4. http://www.microsoft.com/windowsazure/sla/, December 2009.

[40] Microsoft Corporation. Windows Azure Platform Now Generally Available in 21 Countries. http://blogs.msdn.com/windowsazure/archive/2010/02/01/windows-azure-platform-now-generally-available-in-21-countries.aspx, February 2010.

[41] Nimbus. Nimbus EC2 Back-end. http://www.nimbusproject.org/docs/?doc=2.3/faq.html#ec2-backend.

[42] Nimbus. Nimbus Project. http://www.nimbusproject.org/.

[43] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The Eucalyptus Open-Source Cloud-Computing System. In *CCGRID '09: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 124–131, Washington, DC, USA, 2009. IEEE Computer Society.

[44] OpenNebula Project. OpenNebula. http://www.opennebula.org.

[45] O'Reilly et al. Web 2.0 Expo: What is Cloud Computing? Interviews: http://youtube.com/?v=6PNuQHUiV3Q, 2008.

[46] Philipp C. Heckel. Server Virtualization with VMware. http://www.philippheckel.com/files/virtualisation-paper.pdf, November 2008.

[47] Red Hat, Inc. ovirt. http://www.ovirt.org/.

[48] Red Hat, Inc. Red Hat Enterprise Virtualization: Data Sheet. http://www.redhat.com/f/pdf/rhev/DOC049R4-RHEV-Overall-Datasheet.pdf.

[49] Red Hat, Inc. Red Hat Enterprise Virtualization: Hypervisor. http://www.redhat.com/f/pdf/rhev/DOC076-RHEV-Hypervisor.pdf.

[50] Red Hat, Inc. Red Hat Hardware Compatibility List. https://hardware.redhat.com/.

[51] Red Hat, Inc. RHEV Pricing. http://www.redhat.com/f/pdf/rhev/DOC113R8-Pricing-and-Licensing-for-RHEV-for-Servers.pdf.

[52] Red Hat, Inc. Red Hat Enterprise Virtualization for Servers. http://www.redhat.com/virtualization/rhev/server/, April 2010.

[53] salesforce.com. CRM Software & Online CRM System. http://www.salesforce.com/.

[54] Borja Sotomayor, Rubén S. Montero, Ignacio M. Llorente, and Ian Foster. Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing*, 13(5):14–22, September 2009.

[55] Rick Vanover. Comparing Hyper-V R2, vSphere and XenServer 5.5 pros and cons. http://searchsystemschannel.techtarget.com/tip/0,289483,sid99_gci1379509_mem1,00.html, January 2010.

[56] VMware, Inc. VMware vSphere Costs. http://www.vmware.com/vmwarestore/vsphere_purchaseoptions.html.

[57] VMware, Inc. Understanding Full Virtualization, Paravirtualization, and Hardware-Assist. http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf, September 2007.

[58] VMware, Inc. VMware Compatibility Guide. http://www.vmware.com/resources/compatibility/, April 2010.

[59] VMware, Inc. VMware vSphere. http://www.vmware.com/products/vsphere/, April 2010.