Cloud Infrastructure & Applications – CloudIA

Anthony Sulistio, Christoph Reich and Frank Doelitzscher

Department of Computer Science
Hochschule Furtwangen University, Germany
{anthony.sulistio, christoph.reich, frank.doelitzscher}@hs-furtwangen.de

Abstract. The idea behind Cloud Computing is to deliver Infrastructure-as-a-Services and Software-as-a-Service over the Internet on an easy pay-per-use business model. To harness the potentials of Cloud Computing for e-Learning and research purposes, and to small- and medium-sized enterprises, the Hochschule Furtwangen University establishes a new project, called Cloud Infrastructure & Applications (CloudIA). The CloudIA project is a market-oriented cloud infrastructure that leverages different virtualization technologies, by supporting Service-Level Agreements for various service offerings. This paper describes the CloudIA project in details and mentions our early experiences in building a private cloud using an existing infrastructure.

1 Introduction

Although Cloud Computing is a popular trend, it is difficult to get a clear definition of it. Ian et al. [1] discuss the basic concepts of Cloud Computing and show the differences compared to Grid Computing. The key of Cloud Computing lies in its component-based nature, i.e. reusability, substitutability (e.g. alternative implementations, specialized interfaces and runtime component replacements), extensibility, customizability and scalability [2]. In addition, Armbrust et al. [3] give a good overview of Cloud Computing by highlighting obstacles and proposing several opportunities to solve them.

From our point of view, Cloud Computing delivers Infrastructure- and Software-as-a-Service (IaaS and SaaS) on a simple pay-per-use basis. For small- and medium-sized enterprises (SMEs), Cloud Computing enables them to avoid over-provisioning of IT infrastructure and training personnel. Thus, SMEs can take advantage of using a cloud when the IT capacity needs to be increased on the fly. Typically, more resources are needed for services that are available only for a certain period. For example, AF83, a company specializing in social networking and live web solutions, uses Amazon IT infrastructure to deliver a live concert via the web and mobile devices [4]. The concert attracted 7,000 simultaneous users. By using Cloud Computing, AF83 avoids purchasing new hardware for this special event, and delivers this successful event in a short amount of time.

For companies with large IT infrastructure, such as Amazon and Google, becoming a cloud provider allow them to offer their resources to SMEs based on pay-as-you-go and subscription models, respectively. Because not all services need the full resources at the same time for a long period of time, these companies can still use and lease their existing infrastructure with a relatively small cost. Hence, they can reduce the total cost of ownership (TCO) and increase hardware utilization [3].

In a typical university scenario, PC labs and servers are under-utilized during the night and semester breaks. In addition, these resources are on high demands mainly towards the end of a semester. With the aforementioned motivations and scenarios, the Hochschule Furtwangen University (HFU) acknowledges the potential benefits of Cloud Computing. As a result, HFU establishes a new project called Cloud Infrastructure and Application (CloudIA). The targeted users of the CloudIA project are SMEs and HFU staff and students, running e-Science and e-Learning applications. Such use case scenarios are analyzing third party software, software testing and offering a custom-made SaaS. Therefore, in this paper, we introduce our work in building a private cloud in order to support various Quality of Service (QoS) objectives, such as availability, reliability, and security.

The rest of this paper is organized as follows. Section 2 provides some related work. Section 3 explains the CloudIA infrastructure in details. Section 4 shows the potential usage of the CloudIA project, by providing several use case scenarios, whereas Section 5 describes a servlet container application, running on our cloud infrastructure and Amazon. Section 6 mentions our early experiences in building a private cloud. Finally, Section 7 concludes the paper and gives future work.

2 Related Work

As mentioned earlier, companies such as Amazon and Google have become well-known cloud providers. Amazon provides Elastic Compute Cloud (EC2) and Simple Storage Service (S3) that enable users to buy CPU power and data storage, and only pay for their usage. On the other hand, Google offers businesses many of its services and web applications with a flat-rate subscription model. These simple models attract many users, hence, make Cloud Computing more popular and attractive to SMEs.

In academia, the Virtual Computing Laboratory (VCL) [2], developed by North Carolina State University, enables students to reserve and access virtual machines (VMs) with basic a basic image or specific applications environments, such as Matlab and Autodesk. The CloudIA project draws its motivation from VCL by providing students with e-Learning applications, such as a learning management system and a servlet container. The CloudIA project is also targeted towards running e-Science applications and providing customized VMs to SMEs.

The RESERVOIR project [5] proposes a modular and extensible architecture to support federation of cloud providers in a business environment. In the RESERVOIR model, the providers are independent and clearly separated into two roles, i.e. service and infrastructure. The service providers offer SaaS, whereas the infrastructure providers offer IaaS. In contrast, the CloudIA project does not have this distinction, since HFU, as a cloud provider, can offer both IaaS and SaaS. Moreover, the main focus of the CloudIA project is to build a private cloud on an existing infrastructure. Thus, federation of cloud providers is considered as a secondary objective.

Finally, OpenNebula [6], as part of the RESERVOIR project, and Eucalyptus [7] provide users with software to build a private cloud. Hence, OpenNebula and Eucalyptus can be used to deploy, monitor and control VMs on a pool of distributed physical resources. In addition, they provide a seamless interface with Amazon EC2 for scalability purposes. The CloudIA project has similar features to OpenNebula and Eucalyptus, but with added functionalities, such as capacity planning, single sign-on (SSO), and QoS monitoring.

3 The CloudIA Project

To harness the potentials of Cloud Computing for internal usages, and to SMEs, the CloudIA project is established. The main objective of this project is to build a private cloud for the purpose of running e-Science and e-Learning applications. The overview of the CloudIA architecture is shown in Figure 1.

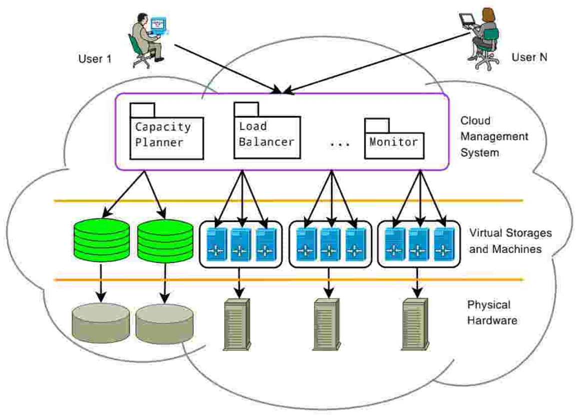

Fig. 1. Overview of the CloudIA architecture.

CloudIA leverages various virtualization technologies, such as Xen [8], KVM [9] and VMware, and supports SLAs as IaaS and SaaS models, as shown in Figure 2. In this figure, the Cloud Management System (CMS) of CloudIA is divided into several layers (such as *User Interface*, *Business*, *System*, and *Resource Interface*) for extensibility and maintainability. However, we incorporate the *Monitoring and Management* and *Security* components across all layers to ensure high reliability and secured services. We describe each layer and component in details next.

3.1 User Interface Layer

This layer provides various access points to users and/or an administrator of the CMS in accessing our cloud system. Such points are described below.

– *Desktop*: allows a direct access by leveraging a Remote Desktop Protocol (RDP).

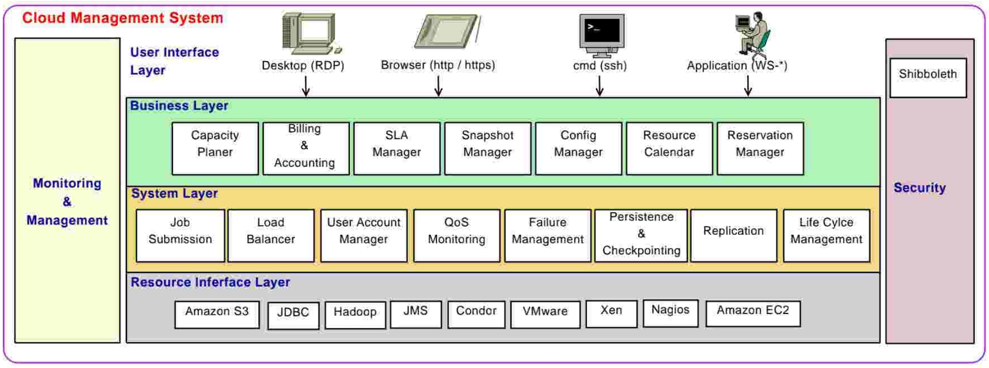

Fig. 2. Cloud Management System of CloudIA.

- *Browser*: gives an access to various Cloud services using web browsers, via secure (*https*) or plain http connections.
- *Command-line interface*: enables an encrypted (*ssh*) connection.
- *Application*: provides Application Programming Interface (API) and uses web technologies (such as WS-* APIs [10]) to interact with other applications.

3.2 Business Layer

This layer aims to regulate resource supply and demand through the use of economy and SLA. In addition, this layer enables users to reserve VMs in advance and manage their personal VMs. Components in this layer are:

- *Capacity Planer*: updates supply and forecasts future demand. In addition, it overbooks resources to reduce risks of unexpected cancellations and no-shows of reservations [11]. It collaborates with the *Reservation Manager* and the *Resource Calendar* components with regards to reservation queries and requests.
- *Billing & Accounting*: calculates costs for using VMs and prices for reserving VMs in advance [12]. In addition, it is responsible for handling user payment.
- *SLA Manager*: handles contracts and negotiations with users regarding to terms and conditions of using our cloud system.
- *Snapshot Manager*: enables installation, data and/or runtime status of applications to be saved, such that users can resume their applications in the future.
- *Config Manager*: creates on-demand VM images according to users' specifications.
- *Resource Calendar*: stores reservations and shows availability of resources in the present and future time.
- *Reservation Manager*: handles reservation queries and requests, by interacting with the *Capacity Planer* and the *Resource Calendar* components [13].

3.3 System Layer

This layer is responsible for daily operation of the CMS, such as submitting jobs, managing user accounts and monitoring QoS. Components in this layer are:

– *Job Submission*: submits users' jobs to resources, according to their SLAs and customized VM images.
– *Load Balancer*: dynamically adjusts load performance of each physical hardware. New algorithms can be incorporated into this component for research purposes.
– *User Account Manager*: manages user accounts, rights and balances. It interacts with Shibboleth [14] for SSO.
– *QoS Monitoring*: monitors jobs and detects any QoS violations.
– *Failure Management*: performs high availability of running VMs, by migrating them into other resources in case of hardware failure. It interacts with the *Load Balancer* and the *Persistence & Checkpointing* components.
– *Persistence & Checkpointing*: saves current status of VMs to enable customer to restart the VM at that point in case of failure.
– *Replication*: performs backup of VM images, services, applications, and other data.
– *Life Cycle Management*: controls and manages the life-cycle of VMs by providing functionalities, such as init, start, stop, move and erase.

3.4 Resource Interface Layer

This layer deals with the physical hardware and hosts many interfaces and Plugins to various virtualization, database, distributed system and other technologies, such as Xen, Java Database Connectivity (JDBC), Hadoop [15], Amazon S3 and EC2 and Nagios [16]. Several interfaces and plugins are described below.

– *JDBC*: provides an independent database connectivity through the use of the Java Database Connectivity (JDBC) API.
– *Hadoop*: enables large files to be stored across multiple machines through the use of the Hadoop File System (HDFS) [15].
– *JMS*: allows distributed applications to communicate with each other by leveraging the Java Message Service (JMS).
– *Condor*: executes compute-intensive jobs on idle resources by interacting with Condor [17].
– *VMware*: provides functionalities of VMware, such as create, run, monitor and migrate VMs.
– *Xen*: provides functionalities of Xen, such as create, run, monitor and migrate VMs.
– *Nagios*: monitors physical resources through the use of Nagios [16].

3.5 Monitoring & Management Component

To ensure the reliability of each layers in the Cloud Management System, a monitoring and management component is needed. Thus, this component allows system administrator to monitor and to initiate activities of each layer, in case of failures, conflicts with SLA objectives, under- or over-utilized resources.

3.6 Security Component

To ensure the privacy, recovery, integrity and security of user data and transactions, a security feature on all layers is required. Part of this requirement is to have a SSO feature, by leveraging the Shibboleth technology. This feature allows a single user credential to access and use various services across multiple layers [18]. Besides the technical solutions, issues in areas such as regulatory compliance and data auditing are important. Therefore, this component is also addressing the aforementioned issues.

4 Use Case Scenarios

To show the potential of the CloudIA project, use case scenarios for SMEs, e-Learning and research purposes are described.

4.1 SME Scenarios

There are many opportunities and advantages for SMEs in using Cloud Computing, such as test new software, evaluate third party applications, increase resources on demand to satisfy seasonal or temporary demand, and offer software to customers as SaaS. SMEs can leverage the CloudIA project for the following scenarios.

– *System and application testing*: Software developers must test their software with different configurations on various operating systems. However, this scenario requires a huge machine farm to satisfy the testing environment. Therefore, SMEs can utilize Cloud Computing by acquiring various resources with different specifications on demand.

– *Analyzing third party software*: Developers and users want to analyze and test software before making a purchase decision. Hence, test installations on a clean operating system with the correct hardware specification are essential. Cloud Computing allows developers and users to try new software by creating new VMs. Another important point to highlight is that existing VMs can easily be removed when they are not needed.

4.2 E-Learning Scenarios

With the CloudIA project, HFU is able to provide the following functionalities:

– *On-demand Learning Management System (LMS)*: HFU leverages the Online Learning and Training (OLAT) project [19] to create a customized e-Learning platform, named Furtwangen E-Learning & Information eXchange (FELIX) [20], as a VM image containing MySQL, Tomcat, Jabber and Apache web server. With this special image, new customers can setup a LMS in the cloud in a matter of minutes, by defining essential configurations, such as a Lightweight Directory Access Protocol (LDAP) host, disk space and logo. In addition, through a federated approach using Sibboleth [21], it is possible to offer FELIX to other universities and institutions.

This on-demand LMS has already been used by HFU for managing alumni information, creating new e-Learning projects and training staff and students.

– *Reservation of a servlet container*: Many programming courses at HFU require students to have their own Java servlet environment for experimentation. However, installation, configuration and management of a servlet container is time consuming. In order to reduce a high learning curve for students, a VM image containing MySQL, Tomcat, PHP, and Apache web server is created. As a result, students can reserve a single VM image exclusively for one semester as part of their courses. With this approach, students can focus more on developing, deploying and testing their applications in a servlet container. More details on this servlet container can be found in Section 5.

4.3 Research Scenarios

The CloudIA project offers IaaS for research purposes, such as:

– *Running compute-intensive applications*: HFU works together with many research and development labs or companies in achieving their research objectives. For example, HFU offers its infrastructure to run parallel jobs of a holographic memory model. As a result, researchers at Thomson, a company specializing in communication and media, can minimize the running time of their simulation model significantly [22].

– *Single System Image*: Many researchers collaborate with other international partners as part of their research project [23]. However, different institutions have their own policies, such as access and security. Thus, the CloudIA project works towards a single system image cloud, by interacting with different middleware and application frameworks, such as Condor [17] and JBoss. As a result, researchers can access several heterogeneous resources and middleware through a common interface.

5 Cloud Application: Servlet Container for e-Learning Purposes

As mentioned earlier, many programming courses at HFU require students to have their own Java servlet environment for experimentation. With a cloud-enabled infrastructure, students are able to reserve VMs, e.g. for 100 hours with pre-defined packages and software for their courses. Since these VMs are running in isolation, IT administrator only need to develop a web front-end embedded in FELIX for authentication purposes. Thus, to access these VMs, students only need to login to FELIX, as shown in Figure 3.

After authentication, the students are able to use servlet container VMs with the following functionalities:

– enables to start and stop servlet container instances, since they can only use these instances for a limited amount of time, e.g. 100 hours.
– saves and deletes snapshots of their work. In this scenario, a snapshot refers to the Tomcat and MySQL configuration details and data, such as war files and database tables.

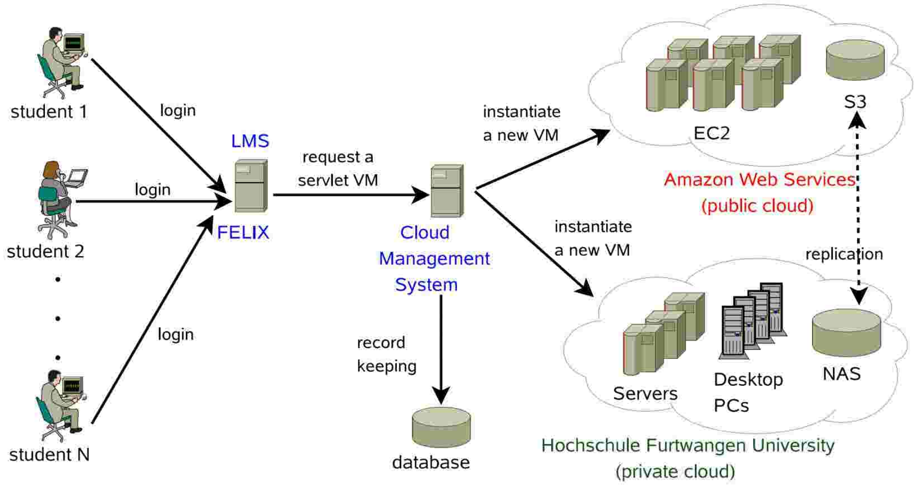

Fig. 3. High level overview of running the servlet container application on the cloud.

– runs a VM with one of the saved snapshots.
– downloads snapshots to a local hard disk. Thus, students only need to send their snapshots for the project submission.

By default, servlet container instances are running on HFU. However, if there are no servers or PCs available, then instances will be running in the public cloud, i.e. using Amazon's small EC2 instances, as shown in Figure 3. Currently, internal IP and dynamic Amazon's public DNS addresses are displayed in FELIX, to allow students to access Tomcat and MySQL through the web. However, these addresses are no longer valid for new instances. In the future, a DNS server will be installed to enable the mapping of the internal IP and dynamic Amazon's DNS of a VM to public names. Therefore, students can access their instances using the same website address throughout semester.

Once the students stop or shutdown their instances, only their snapshots are being saved. The VMs and/or Amazon Machine Images (AMIs) are deleted to save storage. Therefore, these snapshots are replicated between Amazon S3 and our internal network area storage (NAS). In addition, Amazon S3 stores a template AMI for the servlet container with the same basic configurations as the VM hosted on HFU.

By having a servlet container on the cloud, students can start working on their projects, instead of spending time on installing Apache web server, Tomcat and MySQL on their computers. The same scenario can also be applied to a tutor or lecturer in testing and marking the students' work. In addition, students can start a basic VM if they misconfigure the Tomcat and/or MySQL configuration details. Finally, the students can work on their projects using a thin client. Thus, for this application, Cloud Computing provides a fool-proof environment and enhances an e-Learning experience.

6 Early Experiences in Building a Cloud Infrastructure

Currently, many organizations are building their own private clouds from their existing infrastructure [3]. In this section, we describe our experiences and difficulties of building a cloud infrastructure.

6.1 Internal IT Policies

Our university has two departments which are responsible in maintaining and managing IT resources, i.e. IT and Computer Science departments. Each department has its own PC pools, data centers and secured network. The IT department is responsible for managing university-wide resources, whereas the Computer Science department organizes its own resources for teaching and research purposes.

To guarantee a secured network and to prevent unauthorized access, each PC in a computer laboratory has a static IP address, and can only be connected through firewall and MAC-based switch port rules. Thus, with this separation, each department has their own firewall rules and IP subnets. We found that this is obstructive for building a cloud computing infrastructure using heterogeneous resources in the university. We later faced a problem of running out of IP addresses, due to the dynamic creation of virtual instances on the host PCs that belong to same subnet. In addition, these instances could not be connected to the network easily because of unidentified MAC addresses and firewall rules. As the network plays a critical role in the whole concept of Cloud Computing, an organization wanting to create a Private Cloud needs to be wary of the IT policies and practices deployed across internal divisions.

6.2 Technical Incompatibility

To avoid licensing issues and to allow a greater access for customizations, Linux is the preferred choice of host OS for our cloud infrastructure. Since Xen requires a special patched kernel for its host system (*dom0*), this appears to be a major restriction to our existing Linux PCs.

To encourage students to use Linux, Ubuntu is chosen for its ease of use and friendliness. Unfortunately, a Xenified kernel (*dom0*) is no longer supported since Ubuntu 8.10. Therefore, the only solution is to compile an older version of the Linux kernel, e.g. `Linux-2.6.18` (at the time of writing, the latest kernel is `Linux-2.6.28`). However, using an older version of the Linux kernel can cause unexpected problems due to compatibility issues, such as performing software updates and installing Xen on the host OS. For example, `gcc-4.2` is required to make the Xenified Linux kernel, not the latest stable release. As a result, productivity and time were wasted in figuring out the source of the problem and finding the solution. Therefore, technical issues regarding to OS, software and its dependencies need to be addressed when an organization wants to adopt a virtualization technology into its existing IT system.

6.3 Trends in Open Source Virtualization Technologies

In the late 2007, Xen was officially included into the kernel `linux-2.6.23`. In addition, many enterprises and organizations used Xen, e.g. Amazon and Red Hat. As a result, Xen became the standard virtualization technology in Linux. Unfortunately, only `domU` part of Xen, i.e. for running guest VMs, was ported into the kernel.

Due to a long wait of a stable Xenified kernel (*dom0*), several Linux distributions, such as Red Hat and Ubuntu switched to a new virtualization technology, i.e. Kernel-based Virtual Machine (KVM) [9]. Unfortunately, KVM can only be used by the latest processors that support hardware virtualization, such as Intel Core and AMD Athlon. As many organizations prefer to use low-cost hardware for their IT usage, they encounter a dilemma: either upgrade into new hardware or experience the aforementioned technical incompatibility. Thus, it is important to choose a virtualization technology that has a concrete road map and a good long-time support. Moreover, the technology needs to be compatible with the existing PC hardware.

6.4 Co-existence of Virtualization Software

The main objective of the CloudIA project is to better utilize existing resource. In our university's PC labs, students are offered with a variety of software, such as VirtualBox – a desktop virtualization software. Unfortunately, running both KVM and VirtualBox on a single PC is not possible [24]. The only solution is to disable one of these virtualization software. If KVM is disabled, then this PC can not be part of the cloud. Therefore, software dependencies and conflicts need to be addressed and analyzed before installing them.

6.5 Underestimation of Complexity

Typical virtualization scenarios are creating basic VMs and deploying pre-installed VM images. To enable automatic software deployment during the creation and runtime of VMs, we underestimated the technical complexities. As a result, more time is needed than previously anticipated for this task. Therefore, risk management and project plans are essential for a successful and on-time project.

6.6 Running Appropriate Services and Applications on the Cloud

Hosting a particular service or running a legacy application on the Cloud can be counterproductive, i.e. slowing the application down. In addition, it consumes the whole physical resource. Thus, preventing other VMs running on the same resource. For example, our university's time table and room management software, `TimeEdit` [25] was tested on a VM on the HP ESX server. Although the hardware specification meets the requirement of running several virtual instances, `TimeEdit` was constantly consumed the available resource. This is because the underlying database of `TimeEdit` could not be efficiently run on a VM. Hence, it is important to monitor VM consumption periodically, and to run resource-intensive services and applications on dedicated machines.

7 Conclusion and Future Work

The idea behind Cloud Computing is to deliver Infrastructure-as-a-Services (IaaS) and Software-as-a-Service (SaaS) over the Internet on an easy pay-per-use business model. To harness the potentials of Cloud Computing for internal usages, and to small- and medium-sized enterprises, the Hochschule Furtwangen University establishes a Cloud Infrastructure & Application (CloudIA) project – a cloud infrastructure targeted towards e-Science and e-Learning applications. To show the potential of CloudIA, we describe several use case scenarios and early experiences in building a cloud infrastructure.

As for future work, a DNS server to enable the mapping of dynamic IP addresses to public names will be considered. In addition, we will examine the feasibility of moving existing services and e-Learning applications hosted on dedicated servers into the cloud. Finally, a rule-based cloud monitoring component to control virtual machines' connection and interaction with other CloudIA components will be considered.

References

1. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. In: Proceedings of the Grid Computing Environments Workshop (GCE'08), Austin, Texas, USA (Nov. 16 2008)
2. Vouk, M., Averritt, S., Bugaev, M., Kurth, A., Peeler, A., Schaffer, H., Sills, E., Stein, S., Thompson, J.: "Powered by VCL" – Using Virtual Computing Laboratory (VCL) Technology to Power Cloud Computing. In: Proceedings of the 2nd International Conference on the Virtual Computing Initiative (ICVCI'08). (May 16–17 2008)
3. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, University of California at Berkeley (February 2009)
4. Amazon Web Services: AF83 Case Study. http://aws.amazon.com/solutions/case-studies/af83/ (May 2009)
5. Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., Galan, F.: The RESERVOIR Model and Architecture for Open Federated Cloud Computing. IBM Systems Journal (2009)
6. OpenNebula. http://www.opennebula.org/ (May 2009)
7. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The Eucalyptus Open-source Cloud-computing System. In: Proceedings of the 9th IEEE International Symposium on Cluster Computing and the Grid (CCGrid'09), Shanghai, China (May 18–21 2009)
8. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and The Art of Virtualization. In: Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03), New York, USA (Oct. 19–22 2003)
9. Qumranet: KVM: White Paper. http://www.linux-kvm.org/ (2006)
10. Weerawarana, S., Curbera, F., Leymann, F., Ferguson, D., Storey, T.: Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More. Prentice Hall International (2005)
11. Sulistio, A., Kim, K.H., Buyya, R.: Managing Cancellations and No-shows of Reservations with Overbooking to Increase Resource Revenue. In: Proceedings of the 8th International

Symposium on Cluster Computing and the Grid (CCGrid'08), Lyon, France (May 19–22 2008)
12. Sulistio, A., Kim, K.H., Buyya, R.: Using Revenue Management to Determine Pricing of Reservations. In: Proceedings of the 3rd International Conference on e-Science and Grid Computing (e-Science), Bangalore, India (Dec. 10–13 2007)
13. Sulistio, A., Kim, K.H., Buyya, R.: On Incorporating an On-line Strip Packing Algorithm into Elastic Grid Reservation-based Systems. In: Proceedings of the 13th International Conference on Parallel and Distributed Systems (ICPADS), Hsinchu, Taiwan (Dec. 5–7 2007)
14. Shibboleth. http://shibboleth.internet2.edu/ (May 2009)
15. Hadoop. http://hadoop.apache.org/core/ (May 2009)
16. Nagios. http://www.nagios.org/ (May 2009)
17. Thain, D., Tannenbaum, T., Livny, M.: Distributed Computing in Practice: The Condor Experience. Concurrency and Computation: Practice and Experience (CCPE) **17**(2–4) (2005) 323–356
18. Sinnott, R.O., Jiang, J., Watt, J., Ajayi, O.: Shibboleth-based Access to and Usage of Grid Resources. In: Proceedings of the 7th IEEE/ACM International Conference on Grid Computing, Barcelona, Spain (Sep. 28–29 2006)
19. Online Learning And Training (OLAT). http://www.olat.org/ (May 2009)
20. HFU: Furtwangen E-Learning & Information eXchange (FELIX). https://felix2.hs-furtwangen.de (May 2009)
21. Miyata, T., Koga, Y., Madsen, P., Adachi, S.I., Tsuchiya, Y., Sakamoto, Y., Takahashi, K.: A Survey on Identity Management Protocols and Standards. Transactions on Information and Systems **E89-D**(1) (2006) 112–123
22. Eckhardt, M., Garwers, S., Ruebsamen, T., Stoeffelmaier, T., Zehnder, J., Reich, C.: Windows Desktop Grid. Technical Report CRL-2009-01, Hochschule Furtwangen University (January 2009)
23. Delic, K.A., Walker, M.A.: Emergence of the Academic Computing Clouds. Ubiquity **9**(31) (2008) 1–1
24. Ubuntu: KVM FAQ. https://help.ubuntu.com/community/KVM/FAQ (May 2009)
25. TimeEdit. http://www.timeedit.com/ (May 2009)