The novel cloud application technology with virtual platform

Ying-Chuan Chen^a and Yi-Lang Tsai

National Center for High-Performance Computing, Taiwan

Abstract. Virtualization technology of cloud application service is the major research issue in recent years. Our virtual platform can obtain a lot of advantages base on virtualization technology with suitable virtual machine solution that all system resources can deployed quickly and managed competently, and it has some features including the high capacity for reliability, elasticity and customization. In our virtual platform, there are many different application services are provided for cloud service style. For system computing resources, we need to share all of system and software resources using the virtualization technology is presented in related sections, and we proposed virtual and remote application technology to implement innovative, convenient and powerful cloud application service base on virtual platform. For users of our platform, they can easy to use any software and tools on demand with pay per use mode anytime, and they do not have specialized knowledge and ability of install and maintenance.

1 Introduction

Cloud computing [1-2] is a well-known Information Technology (IT) concept, user can convenient access to complete computing resources, which include servers, networks, storage, software, applications and services. All of computing resources are provisioned and released with minimal management effort or service provider interaction. Cloud service is the access to computers and their function through three common deployment styles, such as public cloud, private cloud, and hybrid cloud. Users of the cloud request this access from a set of web service interfaces that operate a pool of computing resources base on the virtualization technology with virtual machines, network devices, storage, etc. Cloud computing solution has many benefits like easy to management, on-demand, customization high scalability, high reliability, pay-per-use, and cost down.

The related features are showed like: First feature is rapid elasticity, services can be elastically deployed and released. In some cases, to scale quickly inward and outward appropriate with demand automatically. For consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time. The second feature is resource pooling, all of computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual computing resources which dynamically allocated and reallocated according to each consumer demand. There is a sense of location independence in that the customer generally has no control and knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (such as country, state, or datacenter). The third

^a Corresponding author : ycc0301@narlabs.org.tw

feature is broad network access, capabilities are available over the network and accessed by standard mechanisms that promote use by heterogeneous thin client devices (including smart phones, tablets and laptops) from anywhere. The fourth feature is measured service, cloud platforms control and optimize resource automatically use by leveraging a metering capability at some level of abstraction appropriate to the type of service like processing, bandwidth, and storage. Resource usage can be controlled and monitored, providing transparency for both the provider and consumer of the utilized service. The fifth feature is on-demand self-service, anytime all of consumers can use physical or virtual computing resources, such as server time and network storage, as needed automatically without requiring human interaction through cloud service providers.

Figure 1. Cloud computing service.

The cloud computing service concept is shown in Figure 1, there are three types of the cloud service models [3-4] showed below:

- Software as a Service (SaaS): Related applications are available remotely. The SaaS services represent a software delivery method that provides access to software and its functions remotely as a web-based service. Providers offer users access to specific application programs controlled and executed on the provider's infrastructure. Among the providers we have Salesforce, Windows Live Mesh, GoogleApps, and facebook. For example, Salesforce is an industry leader in providing online CRM (Customer Relationship Management) cloud services. And Microsoft deliver a SaaS service that called Windows Live Mesh allows files and folders to be shared and synchronized across multiple devices from anywhere.
- Platform as a Service (PaaS): We can develop applications using the services provided. The PaaS services are used for applications, and other development, while providing cloud components to software. Developers gain by PaaS is a framework they can build upon to develop or customize applications. PaaS makes the development, testing, and deployment of applications quick, simple, and cost-effective. All applications and infrastructure are run and managed by services vendor. Google's App Engine is a well-known PaaS service, and it enables users to build Web applications on the same scalable systems that power Google applications.
- Infrastructure as a Service (IaaS): IaaS model are most common services in market, and it provides self-service models for remote accessing, monitoring, and managing datacenter infrastructures. Related equipment of IaaS is provided in the form of virtual machines. The client maintains the applications, runtimes, databases, server software while the supplier maintains the Cloud virtualization, hardware servers, networks and storage components. This service model allows the cloud to operate during high traffic and demanding situations as resources are dynamically increased as they are needed. Typical examples are Microsoft Azure, Amazon Web Services (AWS), Google Compute Engine (GCE) and S3 (Simple Storage Service) where compute and storage infrastructures are open to public access with a utility pricing model.

Virtualization Technology (VT) [5-6] is a very important role, it has the capability to deploy various virtual machines by a hypervisor. The concept for Virtualization Technology showed as Figure 2. A virtual machine (VM) is a software implementation of a machine that executes related

programs like a physical machine. Related physical computing resources are divided as virtual computing resources in virtualization platforms via hypervisor, and those virtual computing resources are assigned to each VM for user’s requirements. Hypervisor of VM is the most important component to control between physical resources and virtual resources. It also providing each Guest OS a set of virtual interfaces. All commands were delivered to hardware layer from virtual layer, which receives results from hypervisor of VMs. There have two different styles of virtualization for Guest OS, such as Full-Virtualization (FV) and Para-Virtualization (PV), which can be combined with Hardware-Assisted Virtualization technology for CPU of Intel or AMD. XEN

Figure 2. The architecture for virtualization technology.

The main goal of our paper is to develop an efficient, scalable and friendly Cloud Application Service on Virtual Platform that is scalable, efficient, and independent of the operating system. The Cloud Application Service adopts open architecture to provide virtualization applications with the heterogeneous operating systems to any client devices. Users can easily obtain virtualization applications in the cloud platform through any web browsers over Internet.

In cloud computing platform, users can use service of SaaS (software as a service) subscriptions instead of traditional software licenses model. In traditional system platform, users need to install operating systems and applications under a granted license. Users may be burdened with many complexes works in terms of software installation, configuration, updating and even troubleshooting issues. In cloud platform, users may access software on demand through the Internet without any installation and maintenance issues. For the system providers, there are two alternative models of making the SaaS service available. The first model is to develop software based on web technologies. This not only requires significant work, but may also encounter compatibility problems with the different browsers. The second model is based on the remote virtual applications that provides a transparent method to deliver software resources based on the virtualization technology. By virtualization technology, all cluster servers and the software which are maintained by the cloud providers and the users just only pay per use. Cloud service providers provide the resources sharing platform for accessing virtual applications, so users develop their programs and execute in the cloud platform.

The related technologies are showed in section II, and framework and implementation of the system is discussed in section III. Conclusion of our paper will be showed in section IV.

2 Technology explanation

The cloud application service is based on cluster architecture with the virtualizing application that is the most important technology. The virtual machine solution is installed with the cluster server. Each application shares the virtual system resources of the cluster platform. The cluster server is the

important infrastructures with high speed network interfaces for the cloud computing technology. The main purpose of the cluster server is to formulate the responses which are sending by the clients, it also executes the high availability and load balancing while the jobs access to make the system reliable.

There have many virtual machine solutions exist for the deployment of open source clouds. Among these, it includes Eucalyptus, OpenNebula and OpenStack. We propose three famous solutions in order to select the most suitable solution to some conditions, such as open source, modularity, popularity, community use and friendly.

2.1 Eucalyptus

Eucalyptus [7] is an open source software platform for implementing Infrastructure as a Service (IaaS) in a cloud computing environment. It allows the installation of a private and hybrid cloud infrastructure. And it is developed in Java and C program language, with a main storage controller walrus and controllers on each node. The network is managed by the component cloud controller and each controller is authenticated by SSH key files and permission to authenticate transactions. There are some features like: (1) Eucalyptus works with various hypervisors including KVM, Xen and VMware. (2) It supports common Linux and Windows OS for virtual machines (VMs). (3) Compatible with Amazon Web Services (AWS) and Simple Storage Service (S3). (4) It is a cloud platform used by Amazon EC2, and the Application Program Interface (API) compatible with Amazon EC2 platform. (5) It can be installed and deployed from source code or DEB and RPM packages. (6) Internal processes communications are secured through SOAP and WS-Security. (7) Multiple clusters can be virtualized as a single cloud. (8) Administrative features such as user and group management and reports. However, Eucalyptus scalability is limited, compared to the massive scalability, and some source code of modules are closed.

2.2 OpenNebula

OpenNebula [8] is another well-known open source virtual machine solution, subject to the requirements of the Apache License version 2. It is developed using C++, Ruby and Shell program language, and it supports authentication via LDAP. The network component is managed manually by the administrator. Storage, the default copy of the virtual machines are accessed through SSH policy. It provides the simple flexible solution for the comprehensive management of virtualized data centers to enable private, public and hybrid IaaS clouds. For data center virtualization management, many users use OpenNebula to manage virtualization data center, consolidate servers, and integrate existing Information Technology (IT) resources including computing, networking and storage. OpenNebula directly integrates with hypervisors, such as KVM, Xen and VMware ESX. It also can complete control all of virtual and physical resources, and providing advanced capability for capacity management, high availability and resource optimization.

There are some features of OpenNebula showed below:

- Openness of the architecture, interfaces, and code.
- Flexibility to fit into any datacenter.
- Simplicity, easy to deploy, operate and use.
- Stability for use in production enterprise-class environments.
- Scalability for large scale infrastructures.
- Lightness for high efficiency.

2.3 OpenStack

The novel OpenStack project is recent and under active development. OpenStack project is licensed under Apache 2.0 license.

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
vm3	CentOS7	• 10.0.0.13	m1.medium	demo	Active	nova	None	Running	10 months, 1 week	Create Snapshot ▾
win7_3	win7_ultimate	• 10.0.0.8 Floating IPs: win7 • 172.24.4.229		-	Active	nova	None	Running	10 months, 4 weeks	Create Snapshot ▾
win7_2	win7_ultimate	• 10.0.0.7 Floating IPs: win7 • 172.24.4.238		-	Shutoff	nova	None	Shut Down	10 months, 4 weeks	Start Instance ▾
win7	win7_ultimate	• 10.0.0.5 Floating IPs: win7 • 172.24.4.228		-	Active	nova	None	Running	10 months, 4 weeks	Create Snapshot ▾
vm1	-	• 10.0.0.9	m1.tiny	-	Shutoff	nova	None	Shut Down	11 months	Start Instance ▾

Figure 3. Web user interface of OpenStack’s instance.

Figure 4. Web user interface of OpenStack’s network.

In Figure 3 and Figure 4 show the web user interface of OpenStack, it is a virtualization platform developed by National Aeronautics and Space Administration (NASA) dedicated to massive infrastructures.

There are three main features showed below:

- **Open:** OpenStack is based on open source concept, the all of code can be modified and adapted if necessary. And there is a validation process for the adoption and development of new standards in the OpenStack project.
- **High scalability:** This solution is already deployed in a lot of companies whose associated data volumes is evaluated in petabytes of distributed architectures and massively scalable over 1 million physical machines, over about 60 million virtual machines and billions of stored objects.
- **High flexibility and compatibility:** This solution supports common virtualization hypervisors of the market: KVM, Xen, QEMU, Hyper-V, ESX, etc.

OpenStack solution is written in python program language and currently implements two control APIs, the EC2 API and Rackspace, and it is completely open source. OpenStack execute suitable drivers to interface with many hypervisors (KVM, Xen, QEMU, Hyper-V, and ESX). This project is dedicated to providing the computer industry with the opportunity to deploy a hosting architecture and massive scalability. It has great potential due to its architecture and community and the support of its partners. According to many-sided survey appears that the best virtualization solution is OpenStack.

Application virtualization [9-12] is layered on top of virtualization technologies. Application virtualization is the separation of an installation of an application from the client computer that is accessing it. Application virtualization can be divided into two types: remote and streaming. Remote applications run on a server. End users view and interact with their applications over a network via a remote display protocol. The remote applications can be completely integrated with the user's desktop so that they appear and behave the same as local applications. The remote applications can be shared with other users, or the application can be running on its own operating system instance on the server. With streaming applications, the virtualized application is executed on the end user's local computer. When an application is requested, components are downloaded to the local computer on demand. Once completely downloaded, a streamed application can work without a network connection. Both forms of application virtualization have benefits from centralized management. Applications can be installed, patched, and upgraded once for an entire environment. Application virtualization typically is combined with application streaming, so the software is not installed in the usual sense. Application virtualization allows software to be run without being installed on the machine and without interacting directly with the original operating system for which it was designed. Application streaming is an on-demand software distribution system. With application streaming, users receive only applications that they request from a master server. In addition, clients are given only the portion of the application that they need to launch the software and use its basic functions [13-15].

The virtual application is different from the virtual desktop sharing. In virtual desktop sharing, a server distributes any screen update. In virtual application sharing, the server distributes screen updates if they belong to the shared application's windows. Client users receive screen updates from the server and send keyboard and mouse events to the server. Application virtualization is also a software technology that encapsulates applications from the underlying operating system on which it is executed. With application virtualization, users can store and manage applications centrally, integrate with third party computer lifecycle management and software distribution systems. Application virtualization also enhances organizations' ability to control access to applications, track usage of virtual applications, and quickly test, deploy and update applications.

Full application virtualization requires a virtualization layer. The application virtualization layers are used to substitute part of the runtime environment which is provided by the operating system. The layer acquires registry operations of virtualized applications and transparently redirects them to a virtualized location. It makes the real application remaining to access the virtual resources. The pieces of the application's code, data, and settings are delivered when first used, and the application streaming are usually delivered over a protocol such as HTTP (Hypertext Transfer Protocol), CIFS (Common Internet File System) or RTSP (Real Time Streaming Protocol) [16-18]. Deploying and

maintaining desktop and application level virtualization poses many challenges, but there are opportunities for improvements at the application and server layers to achieve the high-level of computing performance and scalability [19-20].

3 System design and implementation

In order to implement cloud application service, first we need to build virtual platform base on virtualization technology of OpenStack. Our virtual platform architecture is built using several core components: Identity, Compute, Image, Network and Object. In this section, we will explain in detail these components and we will demonstration the system architecture for our platform in Figure 5.

Figure 5. The system architecture for our platform.

3.1 Identity component

It acts as the key authentication mechanism of our virtual platform across the cloud operating system and can integrate with existing backend directory services like LDAP. It supports multiple forms of authentication including standard username and password credentials, token-based systems and AWS-style (i.e. Amazon Web Services) logins. Additionally, the catalog provides a queryable list of all of the services deployed in a cloud in a single registry. Users and third-party tools can programmatically determine which resources they can access.

3.2 Compute component

It is a management platform that controls the infrastructure to control IaaS clouds. It is designed to manage and automate pools of computer resources and can work with widely available virtualization technologies, as well as bare metal and high-performance computing (HPC) configurations. KVM, Xen, and VMware are available choices for hypervisor technology, as well as Virtual Machine Monitor (VMM), together with Hyper-V and Linux container technology such as LXC. It provides an administrative interface and an API needed for the orchestration of the Cloud. It includes instances management for servers and access control. Compute requires no prerequisite hardware and is completely independent of the hypervisor consists of seven main components. At first, we have the API Server, which acts as the Web front end service control of the hypervisor. Then, we find the Message Queue that implements the mechanism for dispatching the exchanged instructions to facilitate communication. After that, Compute Controller handles the lifecycle of instances, it is

responsible for creating and manipulating virtual servers, while the component Object Store provides storage services.

3.3 Image component

It provides storage services, recording and distributing the images to virtual machine disks. The API of image component provides a standard REST interface for querying information about disk images and lets clients stream the images to new servers. An advance function of this component is the live migration of a running VM, from one physical server to another, without service interruption. Thus, it enables a dynamic and automated self-optimizing virtual platform, allowing hardware maintenance for the underperforming servers without downtimes. Also, it can present detail information about the images, and configure a variation on an image to produce an instance. We can add, delete, share, or duplicate images of VM through image component.

3.4 Network component

It is a system for managing networks and IP addresses. This component ensures the network is not a bottleneck or limiting factor in a cloud deployment, and gives users self-service ability, even over network configurations. It provides network models for different applications or user groups. Standard models include flat networks or VLANs that separate servers and traffic. It also manages IP addresses, allowing for dedicated static IP addresses or DHCP. Floating IP addresses let traffic be dynamically rerouted to any resources in the IT infrastructure, so users can redirect traffic during maintenance or in case of a failure. Besides, users can create their own networks, control traffic, and connect servers and devices to one or more networks. Administrators of platform can use software-defined networking (SDN) technologies like OpenFlow to support high levels of multi-tenancy and massive scale. It provides an extension framework that can deploy and manage additional network services—such as intrusion detection systems (IDS), load balancing, firewalls, and virtual private networks (VPN).

3.5 Dashboard component

It provides administrators and users with a graphical interface to access, provision, and automate deployment of virtual computing resources. The design accommodates third party products and services, such as billing, monitoring, and additional management tools. The dashboard is one of several ways users can interact with virtual machine resources.

Figure 6. The cloud application service architecture.

The cloud application service architecture is showed as Figure 6. The cloud application service system is developed under cluster servers to provide more application services and reduce the software maintenance costs. Applications and shared data can be installed on the cluster server and its hypervisors, and users can request for some specific applications or data on demand from the server. The requested virtual applications will be delivered with dynamic streaming technologies to the client devices.

There are two ways to execute the virtual applications on the client devices: (1) Native mode: The remote control and streaming display software will be installed on the client device to execute the virtual desktops and applications. It is suitable for the homogeneous operating system between the server and clients. (2) Web-based mode: Only need a web browser with Java applet to install on the client device. It is suitable for the heterogeneous operating system between the server and clients.

The virtual application system has a session server to perform in the role of virtualization controller, and to manage user's connections and accountings information. The session server can be implemented with Linux or Windows operating system, and it maintains the virtualization, session and application manager functions. It is responsible for virtualizing the applications and sends them to the proper sessions for the application servers. The session server also communicates with application servers to deliver the virtual application streaming to client devices by RDP technology. Besides, the session server uses the HTTP server to provide the single web-based portal for user login. It also enhances the system security and elasticity by applying the centralized control of applications and files on the server cluster.

When a user requests the application, which installed on the Linux application server, the session server will manage the user account and its session and communicate with Linux application servers. When the Linux application server receives the request from session server, it will start VNC service and make the application virtualize to streaming data, and then delivers the data for the client user.

The virtual application system is composed of several modules: Application Virtualization Manager, User Session Manager and Data Synchronization Manager. The functions of each module are described as follows:

- **Application Virtualization Manager:** This module provides virtualization of applications and software resources. The development of this module is based on the open virtual application software, and designed according to the user requirements. The Full application virtualization also requires a virtualization layer with the operating system. Application virtualization layers replace part of the runtime environment normally provided by the operating system. The layer intercepts all file and registry operations of virtualized applications and transparently redirects them to a virtualized location.
- **User Session Manager:** This module is responsible for managing user connection sessions and authenticating accounts information. The user session begins when the user accesses the virtual desktop or application and ends when the user quits the virtual desktop or application from the web browser. It also plays a role of bridge between application servers and client devices. It applied the SSH (Secure Shell) and HTTP (Hyper Text Transfer Protocol) protocol to provide the single-entry website.
- **Data Synchronization Manager:** This module is responsible for managing the process of establishing consistency among data from cluster servers to client devices and the continuous harmonization of the data over time. Users and project developers can collaborate or develop on a single file without installing any relative application on their own client device. It also backups and shares the user's data to make the system reliable and elastic.

To enable collaborations among multiple virtual machines, an application sharing and migration mechanism will be applied. Through presentation streaming redirection and virtual machine cloning technology, an application can be easily shared or migrated. To realize remote application access in a cloud, RFB protocol is used to transfer the virtual application of a remote virtual machine. The RFB protocol works at the buffer frame layer and supports the remote access to graphical user interfaces, and the mouse or keyboard inputs can be transferred to the remote application, thus achieving a transparent access to the applications. In such a presentation streaming based software delivery mode,

when a client wants to migrate or share an application to another client, the presentation streaming of this application should be redirected and the corresponding virtual machine will be cloned in case of application sharing.

The virtual application system supports both multicast and unicast transmissions. For unicast connections, either UDP or TCP can be used. Since TCP provides reliable communication and flow control, it is more suitable for unicast sessions. Multiple TCP clients sharing a single application may have different bandwidths, so an algorithm which sends the updates at the link speed of each client will be developed. For UDP clients, the system controls the transmission rate because UDP does not provide flow and congestion control. Several simultaneous multicast sessions with different transmission rates can be created at the system. The system can share an application to TCP clients, UDP clients, and several multicasts addresses in the same sharing session.

The application virtualization technology enables applications to be streamed from a centralized location into an isolation environment on the target device where they will execute. The application files, configuration, and settings are transmitted to the target device and the application execution at run time is controlled by the user session manager and virtualization layer. The virtual application system enables collaborative works, and provides an efficient, reliable and elastic service platform for cloud-based computing environment.

Figure 7. The HyperWorks is a cloud application services in our platform.

The cloud application services show as Figure 7 and Figure 8, user can use cloud application like HyperWorks software on our platform by any common web browsers including Internet Explorer, Google Chrome, Mozilla Firefox and Microsoft Edge. In Fig.7, the HyperWorks is a famous Computer-Aided Engineering(CAE) simulation software in the industry. The user interface of ProCAST software is shown in Fig.8, an advanced and complete casting processes simulation tool. In our platform, we can provide many software and tools to achieve user's requirement, even achieve customization function if necessary. For users of platform, they can easy to use any software and tools on demand with pay per use, and don't have related knowledge of install and maintenance anytime.

Figure 8. The ProCAST is a cloud application services in our platform.

4 Conclusion

In recent years, the cloud computing with virtualization technology is the best solution for the increasing efficiency of hardware and software. Virtualization technology acts a very important component to share and control all of virtual system resources in our cloud platform. In our paper, the development of cloud application technology with virtual platform will integrate related cloud computing technologies including IaaS, PaaS and SaaS to provide the innovative application services. Users have not need to worry about the costs of software license and maintenance of servers in our platform. They just only need some easy operations to obtain the various application and software resources on demand everywhere and anytime. In addition, all cloud application services are charge by the pay per use mode. By developing the cloud application technology, more innovative application services of cloud computing can provide to users, and that make the cluster server as a high economic benefits platform.

References

1. D.C. Marinescu, Cloud computing: theory and practice (2017)
2. J.W. Rittinghouse and J.F. Ransome, Cloud computing: implementation, management, and security (2016)
3. P. Mell and T. Grance, The NIST definition of cloud computing (2011)
4. I.A.T. Hashem, et al., Information Systems, **47**, 98-115 (2015)
5. R. Mijumbi, J. Serrat, J.L. Gorricho, N. Bouten, F. De Turck and R. Boutaba, IEEE Communications Surveys & Tutorials, **18**(1), 236-262 (2016)
6. A.S. Ibrahim, J. Hamlyn-Harris and J. Grundy, "Emerging Security Challenges of Cloud Virtual Infrastructure", *Proc. of The Asia Pacific Cloud Workshop (co-located with APSEC 2010)*
7. N. Singh and V. Dhir, International Journal of Advanced Research in Computer Science, **8**(4), 10-13 (2017)
8. A. Vogel, et al., In: *Parallel, Distributed, and Network-Based Processing (PDP)*, 2016 24th Euromicro International Conference on. IEEE, 672-679 (2016)

9. Z. Shen, et al., *International Journal of Modeling, Simulation, and Scientific Computing*, **8**(2) (2017)
10. D. Liu, et al., *In: Audio, Language and Image Processing (ICALIP), 2016 International Conference on. IEEE*, 766-770 (2016)
11. L. Nema, A. Sharma and S. Jain, *International Journal of Engineering Science and Computing*, **6**(6), 7463-7468 (2016)
12. L. Yan, *IEEE Communication Software and Networks ICCSN* (2011)
13. K. Muthu Pandi and K. Somasundaram, *Indian Journal of Science and Technology*, **9**(11), 1-8 (2016)
14. J. Li, J. Huai, C. Hu and Y. Zhu, *Elsevier, Information Sciences*, **180**(17), 3086-3107 (2010)
15. F. Huang and D. J. Beveridge, *Content or File Based Application Virtualization* (U.S. Patent Application No. 15/178, 324., 2017)
16. T. Ilmonen, J. Giulio and M. Petri, *Application sharing between devices in proximity to each other* (U.S. Patent No. 9, 654, 611., 2017)
17. M. Chen, J. F. Liu and J. X. Ying, *Application sharing by dynamic partitioning* (U.S. Patent No. 9, 769, 516., 2017)
18. I. PSARAS, et al., *In: Proceedings of the Workshop on Mobility in the Evolving Internet Architecture, ACM*, 1-6 (2016)
19. Aihua Liang, Zhuo Liu, Limin Xiao, Li Ruan, *In: IEEE International Conference on Future Computer and Communication*, 591-594 (2010)
20. Alexander Gunka, Stepan Seycek and Harald Kühn, *In: Proceedings of the international workshop on Multi-cloud applications and federated clouds, ACM*, 35-42 (2013)