Elastic Cloud Computing Infrastructures in the OpenCirrus^{TM} Testbed implemented via Eucalyptus

Christian Baun and Marcel Kunze

Abstract Cloud computing realizes the advantages and overcomes some restrictions of the grid computing paradigm. Elastic infrastructures can easily be created and managed by cloud users. In order to accelerate the research on data center management and cloud services the OpenCirrus^{TM} research testbed has been started by HP, Intel and Yahoo!. Although commercial cloud offerings are proprietary, Open Source solutions exist in the field of IaaS with Eucalyptus, PaaS with AppScale and at the applications layer with Hadoop MapReduce. This paper examines the I/O performance of cloud computing infrastructures implemented with Eucalyptus in contrast to Amazon S3.

1 Cloud Computing – An upcoming trend in IT

During the last years with the support of public funding grid computing evolved from a computer scientists' field of research to a common working environment for scientific disciplines like physics, medicine, meteorology. At the same time another trend became imminent in the commercial IT sector: Cloud computing [25] aims at consolidating hardware and software resources in large data centers.

Cloud computing realizes the advantages and overcomes some restrictions of the grid computing paradigm [10]. All resources are marketed by providers as a service over the Internet based on a utility model. In the world of cloud computing dynamically scalable (elastic) infrastructures can easily be created and managed by the user. Only the consumed resources are accounted following the pay-as-you-go

---

Christian Baun
Steinbuch Centre for Computing, Forschungszentrum Karlsruhe GmbH, Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany. e-mail: baun@iwr.fzk.de

Marcel Kunze
Steinbuch Centre for Computing, Forschungszentrum Karlsruhe GmbH, Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany. e-mail: kunze@iwr.fzk.de

paradigm. Many different cloud definitions exist and according to our understanding we want to phrase the following short and concise definition:

> Building on compute and storage virtualization, and leveraging the modern Web, cloud computing provides scalable, network-centric, abstracted IT infrastructure, platforms, and applications as on-demand services that are billed by consumption. [6]

Based on this definition we compare grids and clouds. The actual situation is that most cloud infrastructures consist of one or few data centers with heterogeneous or homogeneous resources under central control. A grid consists of geographically distributed, heterogeneous resources without no central control. The primary user groups for cloud computing is industry and business, while grids are prominently in use for research and are operated by academic organizations.

Cloud infrastructures are well suited for generic applications while grids are used for special application domains like high energy physics. Clouds are run by service providers, have commercial busines models (e.g. Pay-as-you-go) and use proprietary middleware systems. Grids are mostly publicly funded. Therefore the usage is for free and the middleware systems are Open Source.

A strong advantage of clouds is the more comfortable usability, as the ownership of resources is granted to the service consumer. While the services in grid infrastructures are mostly handcrafted, clouds provide fully automated services. For this reason cloud computing forms a major step towards the industrialization of IT.

1.1 Everything as a service

When talking about cloud computing it must be kept in mind that different types of cloud services exist (see Figure 1).

- **Infrastructure as a Service** (IaaS) implements an abstract view towards the hardware (servers, network,...) and allows to run virtual instances of servers without the need to directly access the bare metal. Popular IaaS offerings are 3tera [1], Amazon EC2 [2], Flexiscale [16], GoGrid [19], Joyent [24] and Todo flexIT [34].
- **Platform as a Service** (PaaS) takes the level of abstraction further. PaaS appears as a virtual appliance and makes it simple to scale from a single server to many. Here, the user has no need to worry about the operating system and related application software packages. Popular PaaS offerings are Bungee Connect [8], Force.com [17], AppEngine [21], Microsoft Azure [26] and Zoho Creator [36].
- **Software as a Service** (SaaS) provides enterprise quality software (complete applications) to be consumed as a utility. Popular SaaS offerings are Clarizen [9], Gliffy [18], Google Apps [20], Salesforce.com [32], SlideRocket [33], Oracle SaaS Platform [29] and Zoho [37].

Figure 1 also includes an overview of a selection of the most popular IaaS, PaaS and SaaS cloud services.

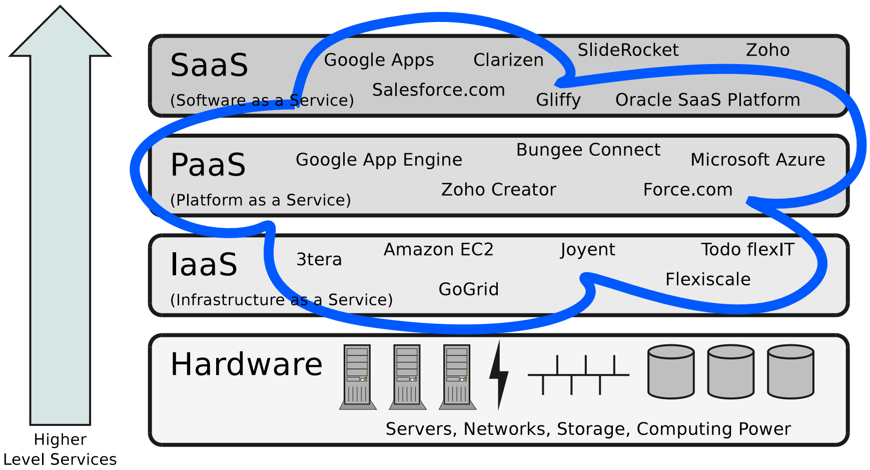

Fig. 1 Different types of cloud services

Cloud computing has the potential to radically change the way IT services are implemented and managed. Project and business funds can be spent to support the core business rather than spending it for IT infrastructure. As they have resource ownership, cloud users are free to run the operating systems, infrastructures, applications and programming languages of their choice. The flexibility of cloud computing has its origin in the combination of virtualization technologies with web services.

2 OpenCirrus™

In July 2008 the OpenCirrus™ project [27] was announced by HP, Intel and Yahoo!. OpenCirrus™ aims to build an open, internet-scale global testbed for cloud computing research focussing on data center management, cloud services, systems and application level research [28]. OpenCirrus™ is a loose federation of three sponsors which are HP Labs, Intel Research and Yahoo!. At the time being there are three academic partners: The University of Illinois at Urbana-Champaign (UIUC), the Singapore Infocomm Development Authority (IDA) and the Karlsruhe Institute of Technology (KIT).

OpenCirrus™ consists of six sites initially, hosted by the sponsors and partners, each equipped with up to 1000 CPU cores and 1 Petabyte of data store initially.

2.1 OpenCirrus™ Blueprint

The foundation of the OpenCirrus™ research testbed is formed by the Physical Resource Sets (PRS). These provide logical mini-datacenters to the researchers and isolate the experiments from each other (see Figure 2). Inside a PRS ensembles of physical nodes are allocated and isolated via vlans using already existing software like Emulab, a network emulation testbed from the University of Utah and HP Opsware, a tool for server provisioning, configuration and management.

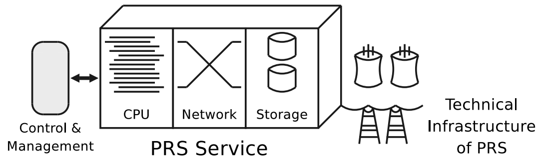

Fig. 2 Physical Resource Sets - The basis of OpenCirrus™ IT Infrastructure

The PRS are the basis to implement Virtual Resource Sets (VRS). The VRS abstract from physical resources by the introduction of a virtualization layer. The virtualization concept applies to all IT aspects like CPU, storage, networks and applications. The main advantages of VRS is the potential to create IT services exactly fitting customers varying needs. IT services can be deployed on demand by automated resource management. Service levels can be easily guaranteed and live migration of services is made possible. As a consequence capital expenditures and operational expenditures are both reduced.

The VRS are implementing IaaS because they provide compute, storage and networking services.

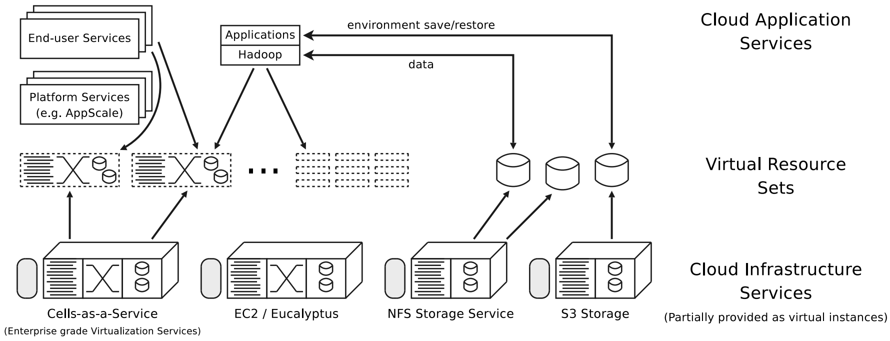

Fig. 3 OpenCirrus™ Blueprint

Figure 3 shows the OpenCirrus™ blueprint with the cloud infrastructure services based on the PRS, the VRS and on top the end-user services, platform services and applications.

For gaining access to OpenCirrus^{TM} project principal investigators apply to each site separately with their contact names, email addresses, and web links for applications via the OpenCirrus^{TM} website. Each OpenCirrus^{TM} site decides which users and projects are accepted. It is planned to have a global sign on for all sites. Users will be able to login to each OpenCirrus^{TM} site for which they are authorized using the same login and password. Three different types of users can use OpenCirrus^{TM} sites:

- Individual principal investigators from academic research groups.
- Industry researchers from the OpenCirrus^{TM} partners.
- Industry researchers who have a customer relationship with the OpenCirrus^{TM} partners.

The majority of users will be academic researchers and researchers who collaborate with the OpenCirrus^{TM} partners. There will be a few carefully chosen users who are industry researchers with a customer relationship with an OpenCirrus^{TM} partner.

The OpenCirrus^{TM} project is seeking research in the areas of datacenter federation, datacenter management, web services, data-intensive applications and Hadoop MapReduce applications. Projects following traditional HPC application development and production applications that just need lots of cycles are not of primary interest.

OpenCirrus^{TM} strongly differs from other cloud computing testbeds because it supports both, system and application level research. In contrast to commonly used cloud infrastructures like Amazon Elastic Compute Cloud (EC2) [2], Amazon Simple Storage Service (S3) [3] and Google AppEngine [21], all software layers and the hardware itself can be accessed and adapted by the OpenCirrus^{TM} researchers. Intel platform features like Intel Data Center Management Interface (DCMI) and Node Manager (NM) that support cloud computing could also be utilized.

3 Cloud application services in OpenCirrus^{TM}

A subset of the cloud application services that will be implemented and used by KIT in the OpenCirrus^{TM} research testbed is Hadoop for MapReduce applications, the cloud computing software infrastructure Eucalyptus and the platform service AppScale.

3.1 Hadoop

One of the main applications of OpenCirrus^{TM} is Hadoop [23]. Hadoop is an Open Source Java framework developed by the Apache Software Foundation and sponsored by Yahoo!. Its intent is to reproduce the Google proprietary software infrastructure for scaling Web applications. It provides MapReduce, a distributed file system called Hadoop Distributed File System (HDFS) that is inspired by the Google File System and a parallel database (HBase). MapReduce as a parallel programming model supports distributed computing on large data sets, e.g. petabyte of data across many objects can be sorted in only a few hours.

3.2 Eucalyptus

As of today, commercial cloud offerings (see Figure 1) are proprietary and usually not open for cloud systems research and development.

However, since December 2008, with the introduction of Eucalyptus the situation has chanced dramatically. Eucalyptus [13] is an Open Source software developed at University of California, Santa Barbara, implementing cloud computing on university compute clusters. EUCALYPTUS stands for Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems. It implements IaaS while giving the user the ability to run and control virtual machine instances via Xen or KVM deployed across a variety of physical resources [14].

The interface is fully compatible with Amazon EC2 and Eucalyptus includes Walrus, a basic implementation of the Amazon S3 interface. For short, it can be said that Eucalyptus is the Open Source implementation of the Amazon Web Services interfaces.

Eucalyptus is able to leverage the same tools known to work with Amazon EC2 and S3. The project has thus the potential to help establish an open cloud computing infrastructure standard. The main components of Eucalyptus are the Cloud Controller (CLC), Cluster Controller (CC) and Node Controller (NC) [15].

The NC runs on every node in the cloud as well as a Xen-Hypervisor [35] or KVM [22]. The NC provides information about free resources to the CC. The CC schedules the distribution of virtual machines to the NC and collects (free) resource information. The CLC collects resource information from the CC and operates like a meta-scheduler in the cloud. Figure 4 shows the structure of Eucalyptus including CLC, CC and NC.

For preliminary testing two Eucalyptus R&D cloud installations have been set up at KIT. KIT Eucalyptus R&D cloud I running Eucalyptus 1.4 and KIT Eucalyptus R&D cloud II running Eucalyptus 1.5 with AppScale 1.1.

- KIT Eucalyptus R&D cloud I
  - 2x IBM Blade LS20 (2x Single Core Opteron at 2.4 GHz, 4 GB RAM)
  - 2x IBM Blade HS21 (2x Dual Core Xeon at 2.33 GHz, 16 GB RAM)
- KIT Eucalyptus R&D cloud II
  - 5x HP Blade ProLiant BL2x220c. Each Blade includes two Server Nodes (2x Intel Quad-Core Xeon at 2.33 GHz, 16 GB RAM)

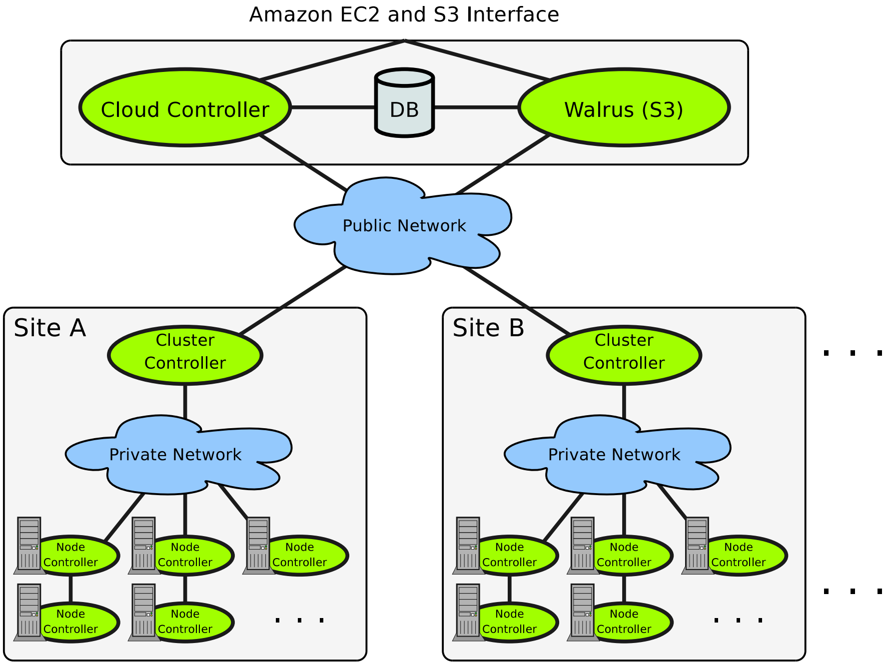

Fig. 4 Structure of Eucalyptus

3.2.1 Storage Performance of Eucalyptus.

In order to compare the storage performance of Amazon S3 and Eucalyptus, the benchmark tool `Bonnie++` [7] was used. This tool measures the rate of sequential output and sequential input. The measurements can be seen in Figure 5 and Figure 6. The measurements are in detail:

- Sequential Output
  - Per-Character: file is written using `putc()`
  - Block: file is written using `write()`
  - Rewrite: `read()` and `write()`
- Sequential Input
  - Per-Character: file is read using `getc()`
  - Block: file is written using `read()`

The RAM of the Eucalyptus NCs was reduced to overcome memory caching effects. Eucalyptus' storage performance depends on the storage subsystem available. For these tests, the write performance of Eucalyptus, using a modern SAS harddisk with 10000 RPM (revolutions per minute) is faster than Amazon S3. The performance for read in contrast is faster at the Amazon sites.

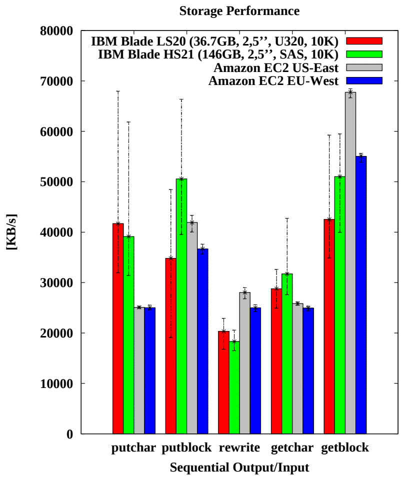

Fig. 5 Storage Performance of Amazon S3 and Eucalyptus

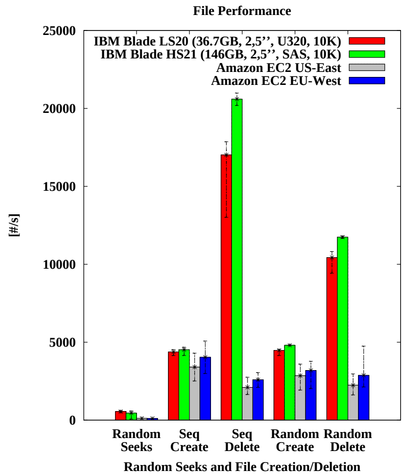

Fig. 6 Performance of Random Seeks and File Creation/Deletion for Amazon S3 and Eucalyptus

`Bonnie++` also measures random seeks and file creation (see Figure 6). Both is faster with Eucalyptus. A possible explanation for these findings is the close distance between the Eucalyptus NCs and the data stored locally. It is unknown whether data stored by Amazon S3 is located near or far the EC2 instances.

3.3 AppScale

A platform service used at the OpenCirrus^{TM} KIT site is AppScale [4]. AppScale is an Open Source implementation of the Google AppEngine development platform. AppScale executes automatically and transparently over cloud infrastructures such as Eucalyptus and provides a PaaS cloud service that allows users to deploy, test, debug, measure, and monitor original Google AppEngine applications.

Figure 7 shows an overview of the AppScale design. It consists of an AppLoadBalancer, one or more AppServers, a Database Master and one or more Database Slaves.

The Google AppEngine application users interact with the AppServers. Developers deploy Google AppEngine applications and AppScale itself through the head node that includes the AppLoadBalancer using the AppScale Tools. The AppServers interact with the Database Master of the AppScale system via HTTPS. [5]

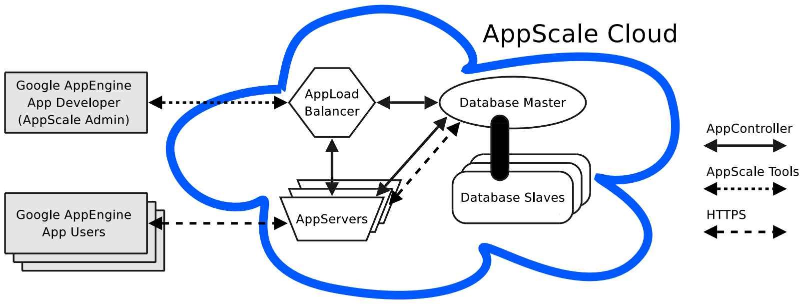

Fig. 7 Overview of the AppScale design [5]

4 Further Steps

At KIT we intend to integrate CernVM with Eucalyptus. CernVM is a system to create virtual software appliances. It offers demand-driven and user friendly creation of virtual machines for various operating systems and applications.

Another plan is to adapt g-Eclipse for use with Eucalyptus. g-Eclipse was an European R&D project (07/2006 - 12/2008) that developed a user-friendly graphical client to manageg grid resources. Several grid middleware systems (gLite, GRIA, GT2, GT4) and common cloud infrastructures (EC2, S3) are supported. g-Eclipse supports the following roles with their corresponding tasks:

- User: Data Management, Job Management, Visualization...
- Operator: Site Administration, User Administration, Job Management...
- Developer: Code Applications, Compile Apps, Debug Apps...

Another intention is to improve the interoperability of EC2/S3 tools (e.g. ElasticFox [12], S3Fox [30], ElasticDrive [11], S3tools [31]) with Eucalyptus and transfer grid services into the cloud (Grid as a Service).

5 Conclusion

Cloud computing allows for flexible and elastic resource provisioning and the large economy of scale makes it attractive for both, academia and business. Using a high degree of automation, cloud computing seems to path the way from manufacture towards the industrialization of IT, thus reducing both, CapEx and OpEx.

OpenCirrus™ offers interesting R&D opportunities for cloud systems and application development.

With Eucalyptus and AppScale, two Open Source implementations of the perhaps most popular IaaS and PaaS offerings are available, representing a first step towards the creation of cloud standards.

References

1. 3tera. http://www.3tera.com
2. Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/
3. Amazon Simple Storage Service (Amazon S3). http://aws.amazon.com/s3/
4. AppScale. http://code.google.com/p/appscale/
5. Navraj Chohan, Chris Bunch, Sydney Pang, Chandra Krintz, Nagy Mostafa, Sunil Soman, Rich Wolski. *AppScale Design and Implementation*. January 2009.
6. Christian Baun, Marcel Kunze, Jens Nimis and Stefan Tai. *Cloud Computing: Web-basierte dynamische IT-Services*. Sprinter. September 2009.
7. Bonnie++. http://sourceforge.net/projects/bonnie/
8. Bungee Connect. http://www.bungeeconnect.com
9. Clarizen. http://www.clarizen.com
10. Marc-Elian Bégin. *An EGEE Comparative Study: Grids and Clouds – Evolution or Revolution*. CERN. June 2008. https://edms.cern.ch/file/925013/4/EGEE-Grid-Cloud-v1_2.pdf
11. ElasticDrive. http://www.elasticdrive.com
12. ElasticFox. http://s3.amazonaws.com/ec2-downloads/elasticfox.xpi
13. Eucalyptus. http://open.eucalyptus.com
14. Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff and Dmitrii Zagorodnov. *The Eucalyptus Open-source Cloud-computing System*. October 2008.
15. Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff and Dmitrii Zagorodnov. *Eucalyptus: A Technical Report on an Elastic Utility Computing Archietcture Linking Your Programs to Useful Systems*. August 2008.
16. Flexiscale. http://www.flexiscale.com
17. Force.com. http://www.salesforce.com/force/
18. Gliffy. http://www.gliffy.com
19. GoGrid. http://www.gogrid.com
20. Google Apps. http://www.google.com/apps/
21. Google AppEngine. http://code.google.com/appengine/
22. Kernel Based Virtual Machine. http://www.linux-kvm.org
23. Hadoop Map/Reduce. http://wiki.apache.org/hadoop/ProjectDescription
24. Joyent. http://www.joyent.com
25. Rajkumar Buyya, Chee Shin Yeo and Srikumar Venugopal. *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*. 2008
26. Windows Azure Platform. http://www.microsoft.com/azure/
27. OpenCirrus™ Cloud Computing Research Testbed. https://opencirrus.org
28. Roy Campbell, Indranil Gupta, Michael Heath, Steve Ko, Michael Kozuch, Marcel Kunze, Thomas Kwan, Kevin Lai, Hing Yan Lee, Martha Lyons, Dejan Milojicic, David O'Hallaron, and Yeng Chai Soh. *OpenCirrus™ Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research*. 2009
29. Oracle SaaS Platform. http://www.oracle.com/us/technologies/saas/
30. S3Fox Organizer. http://www.s3fox.net
31. S3tools. http://s3tools.org
32. Salesforce.com. http://www.salesforce.com
33. SlideRocket. http://www.sliderocket.com
34. Todo flexIT. http://www.todo.de/Produkte/flexIT.php
35. Xen Hypervisor. http://www.xen.org
36. Zoho Creator. http://creator.zoho.com
37. Zoho. http://www.zoho.com