

Cloud-Infrastrukturen mit Linux und Eucalyptus aufbauen

LinuxTag 2009 (Berlin)

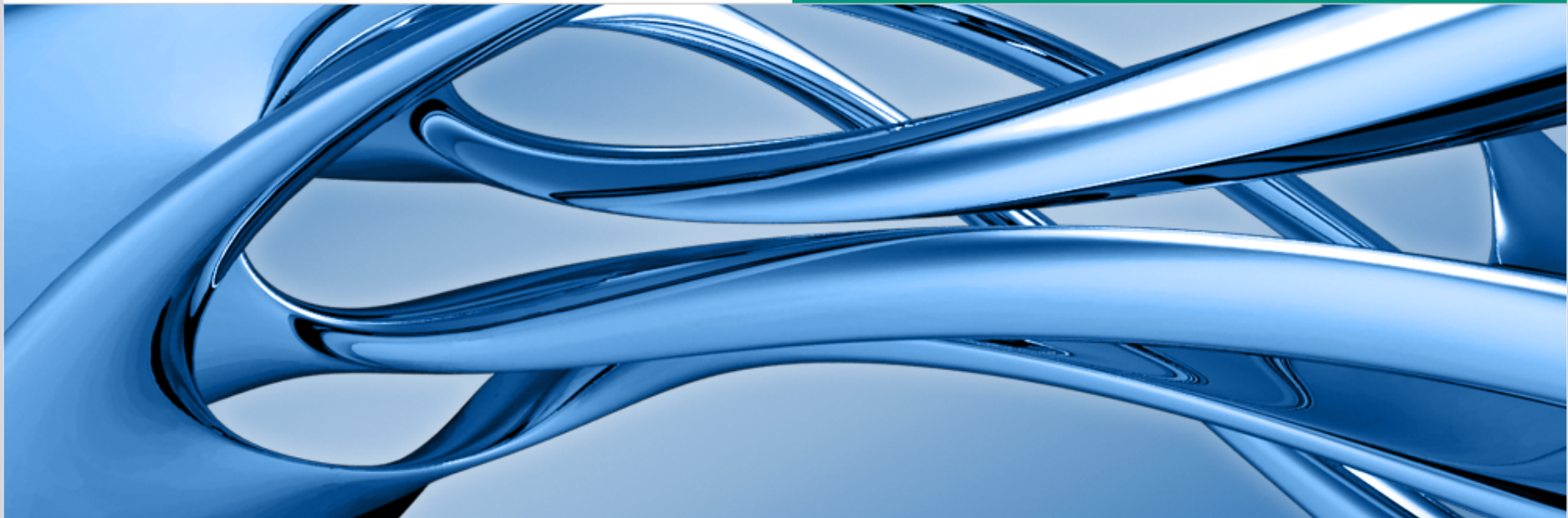
Christian Baun



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft



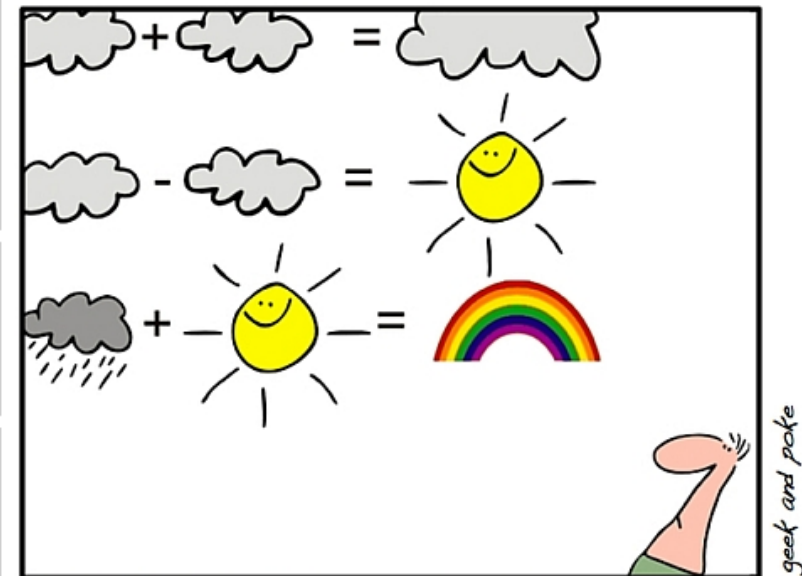
Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825



Cloud-Computing auf dem LinuxTag 2009

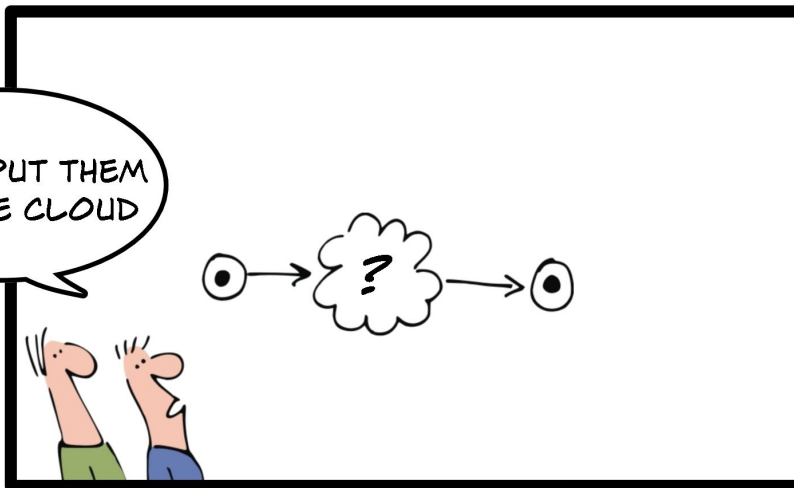
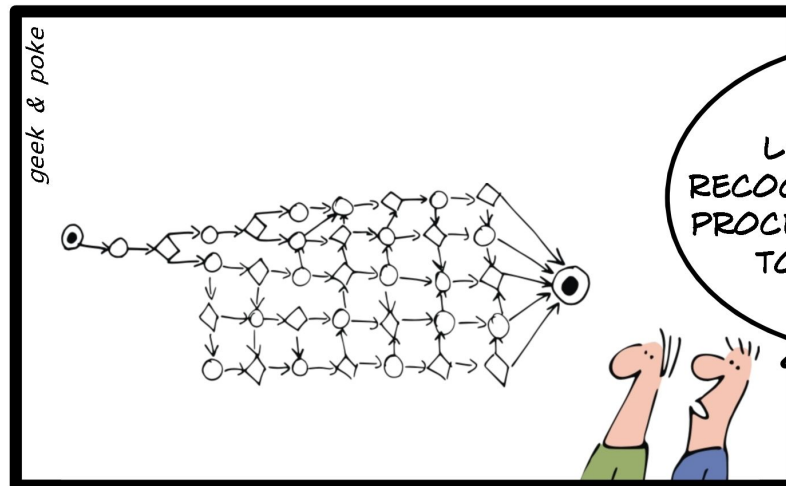
Zeit	London	Berlin I	Berlin II
10:00	O OOoPortable -  OpenOffice.org für die Hosentasche Andreas Mantke (OpenOffice.org)	S Centralized  OpenVPN Thorsten Robers (OpenSource Training Ralf Spenneberg)	C Freie Software  im Stundenhotel - Software-Architektur und Einsatzmöglichkeiten von Amazon EC2 & Co Nathanael Obermayer (selbständiger Berater)
10:30	o Zentrale Konfiguration von OpenOffice.org im Netzwerk - ein Praxisbeispiel Florian Effenberger (OpenOffice.org)	e	l
11:00	f Projekt Renaissance -  Auf dem Weg zu einem neuen Benutzungskonzept für OpenOffice.org Christoph Noack (OpenOffice.org)	c CAcert was war, was ist, was sein wird Henrik Heigl (CAcert Inc. / Fedora Project EMEA)	o The Open Cloud:  Open Source and Cloud Computing Converge Mitch Lieberman (SugarCRM Inc.)
11:30	f	i	u
12:00	f At the development crossroad : the OpenOffice.org case ! Sophie Gautier (LINAGORA)	u OpenVAS -  Neue Wege im Vulnerability Assessment Jan-Oliver Wagner (Intevation GmbH)	d Zimory: a Marketplace for Cloudcomputing Thomas Fricke (Zimory)
12:30	u	g	g Cloud-Infrastrukturen mit Linux und Eucalyptus aufbauen Christian Baun (Forschungszentrum Karlsruhe GmbH)
13:00	g	h	h

- Allgemein großes Interesse am Cloud-Computing
- Hype-Thema seit Ende 2008
- Was steckt dahinter?



**SIMPLY EXPLAINED - PART 17:
CLOUD COMPUTING**

Warum Cloud Computing?



LET THE CLOUDS MAKE YOUR LIFE EASIER

- **Cloud-Computing soll:**
 - IT revolutionieren
 - Komplexität verringern
 - IT flexibilisieren (elastic)
 - Kosten senken
 - Einfachen Ressourcenzugriff bieten
 - Benutzerwünsche erfüllen
 - Benutzer emanzipieren
 - ...
- **Wir erinnern uns: GRID !!!**

Definitionen: Cloud- und Grid-Computing

- **Cloud Computing is on-demand access to virtualized IT resources that are sourced inside or outside of a data center, scalable, shared by others, simple to use, paid for via subscription or as you go and accessible over the web.**

Dr. Behrend Freese (Zimory GmbH)

- **A computing Cloud is a set of network enabled on demand IT services, scalable and QoS guaranteed, which could be accessed in a simple and pervasive way.**

Dr. Marcel Kunze (SCC/KIT)

- **Grid computing is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.**

Ian Foster (Argonne National Laboratory)

- **A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.**

Ian Foster & Carl Kesselman

Cloud-Comuting = Grid-Computing ?!

Cloud-Computing

Ein oder wenige Rechenzentren mit heterogenen oder homogenen Ressourcen unter zentraler Kontrolle

Virtualisierte Ressourcen

Nutzung primär durch Industrie und Wirtschaft
(große Chance für Startups)

Verbrauchsabhängige Abrechnung
(Pay-as-you-go)

Benutzerfreundlich

Grid-Computing

Geographisch verteilte, heterogener Ressourcen ohne zentrale Kontrollinstanz nach dem Prinzip der Virtuellen Organisationen

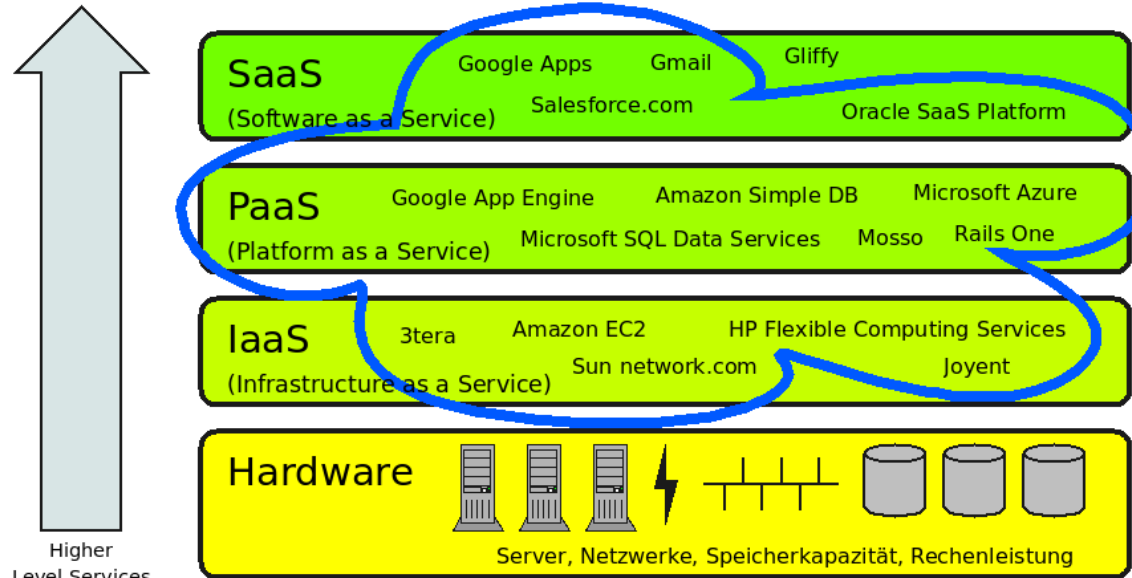
Physische Ressourcen

Nutzung primär durch Forschungs- und Bildungseinrichtungen
(wissenschaftliche Großprojekte wie LHC)

Förderung durch die öffentliche Hand.
Freie Nutzung nach Zustimmung durch die Ressourcenbetreiber

Lange Einarbeitungszeit notwendig

Drei Kategorien von Cloud-Systemen



■ SaaS

- Applikationen werden durch Dienstleister betrieben
- Anwender kümmern sich nicht um Installationen, Wartung (Administration) oder das Einspielen von Updates

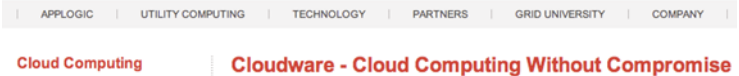
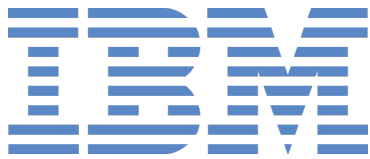
■ PaaS

- Erscheint als ein großer Rechner und macht es einfach zu skalieren
- Anwender müssen sich keine Gedanken um das Betriebssystem oder Systemsoftware machen

■ IaaS

- Abstrahiert die Hardware. Ermöglicht das Starten virtueller Instanzen

Kommerzielle Cloud Anbieter (kleine Auswahl)



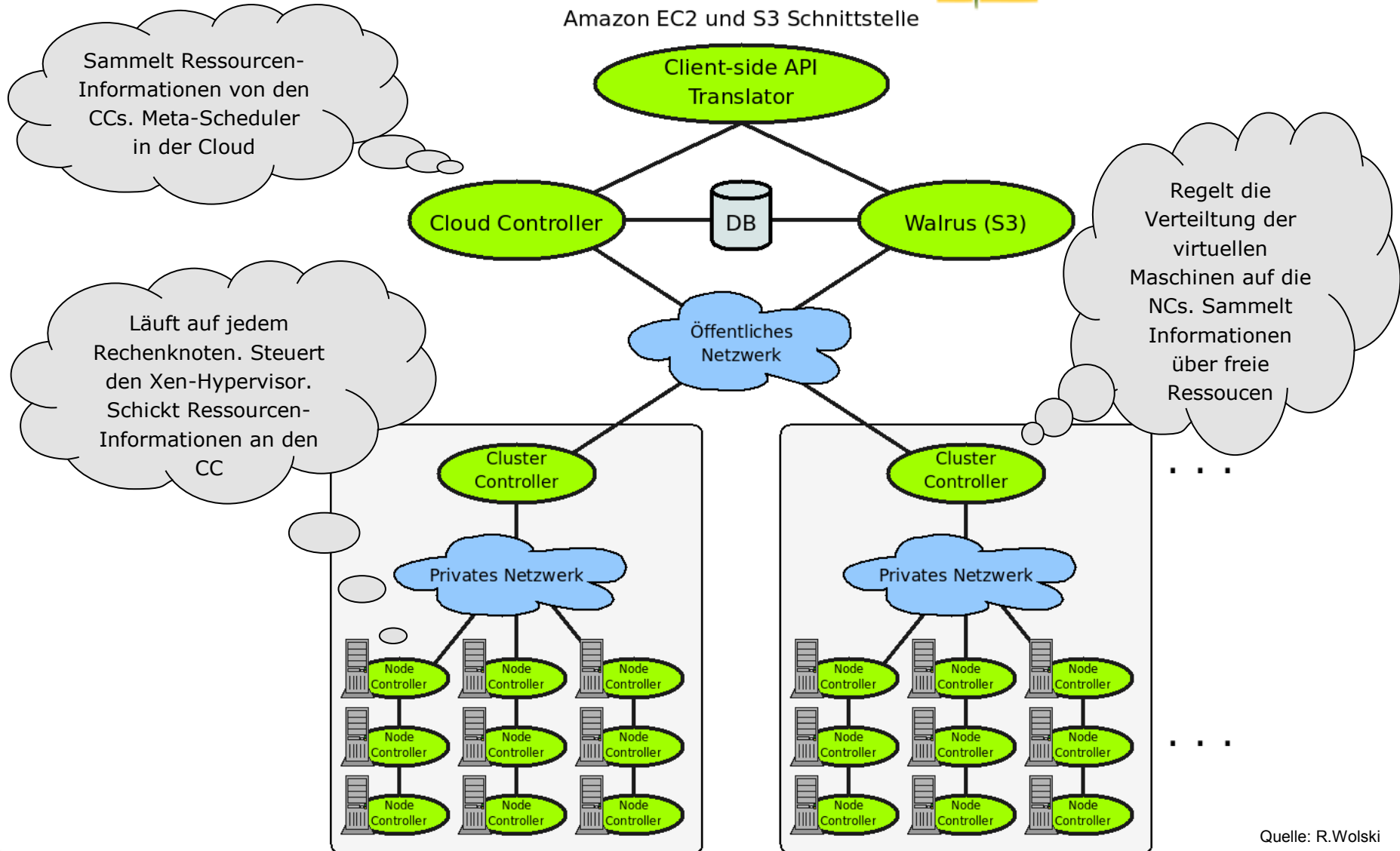
- Problem: Kommerzielle Cloud-Computing Angebote sind meist proprietär und damit ungeeignet für den Aufbau eigener Cloud-Infrastrukturen (Private Cloud)



- **EUCALYPTUS - Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems**
- **Open-Source Software um Cloud-Infrastrukturen (Private Cloud) auf Clustern aufzubauen**
- **Entwickelt an der UC Santa Barbara**
- **Implementiert IaaS. Ermöglicht die Ausführung und Kontrolle virtueller Instanzen auf Basis von Xen oder KVM auf verschiedenen physischen Ressourcen**
- **Die Schnittstelle von Eucalyptus ist kompatibel zu Amazon EC2 (Elastic Compute Cloud)**
- **Enthält einen Speicher-Dienst („Walrus“), dessen Schnittstelle kompatibel zu Amazon S3 (Simple Storage Service) und Amazon EBS (Elastic Blockstore) ist**
- **Eucalyptus ist ein wichtiger Schritt hin zur Etablierung eines Standards für Cloud-Computing-Infrastrukturen**



Amazon EC2 und S3 Schnittstelle



Quelle: R.Wolski

- **Cloud-Installation I (Eucalyptus 1.4)**
 - Produktionsumgebung. Läuft stabil
 - Plattform für Performance-Tests
 - 2x IBM Blade LS20
 - Dual Core Opteron (2,4GHz)
 - 4GB Hauptspeicher
 - 2x IBM Blade HS21
 - Dual Core Xeon (2,33GHz)
 - 16GB Hauptspeicher

- **Cloud-Installation II (Eucalyptus 1.5.1)**
 - Noch im Aufbau
 - 5x HP Blade ProLiant BL2x220c
 - Pro Blade: 2 Server
 - 2x Intel Quad-Core Xeon (2,33GHz)
 - 16GB Hauptspeicher

- **Binärpakete existieren für CentOS, openSUSE, Debian und Ubuntu**
<http://open.eucalyptus.com/downloads>
http://open.eucalyptus.com/wiki/EucalyptusAdministratorGuide_v1.5
- **Eucalyptus kann auch aus den Quellen installiert werden**
http://open.eucalyptus.com/wiki/EucalyptusSourceCodeInstallation_v1.5
- **Anforderung: ≥ 1 Rechner mit Linux und Xen-Hypervisor oder Kernel-based Virtual Machine (KVM)**
 - Für KVM ist eine *moderne* CPU mit AMD-V (Pacifica) oder Intel VT (Vanderpool) notwendig
- **Amazon EC2 Kommandozeilen-Tools (zur Steuerung)**
 - **ec2-api-tools-1.3-30349**
 - **ec2-ami-tools-1.3-26357**

■ Hostname des Frontends, freie Ressourcen und verfügbare NCs ausgeben

```
# ec2-describe-availability-zones verbose
AVAILABILITYZONE      Cluster1    iwrcgblade11
AVAILABILITYZONE      |- vm types      free / max      cpu      ram      disk
AVAILABILITYZONE      |- m1.small       0020 / 0024      1        128       10
AVAILABILITYZONE      |- c1.medium      0020 / 0024      1        256       10
AVAILABILITYZONE      |- m1.large       0008 / 0012      2        512       10
AVAILABILITYZONE      |- m1.xlarge      0008 / 0012      2        1024      20
AVAILABILITYZONE      |- c1.xlarge      0002 / 0006      4        2048      20
AVAILABILITYZONE      |- iwrcgblade11   certs[cc=false,nc=false]
                        @ Thu May 14 22:16:23 CEST 2009
AVAILABILITYZONE      |- iwrcgblade12   certs[cc=false,nc=false]
                        @ Thu May 14 22:16:23 CEST 2009
AVAILABILITYZONE      |- iwrcgblade13   certs[cc=false,nc=false]
                        @ Thu May 14 22:16:23 CEST 2009
AVAILABILITYZONE      |- iwrcgblade30   certs[cc=false,nc=false]
                        @ Thu May 14 22:16:23 CEST 2009
```

■ Dateisystem-Image registrieren:

```
# ec2-bundle-image -i debian5.img  
# ec2-upload-bundle -b debian5 -m /tmp/debian5.img.manifest.xml  
# ec2-register debian5/debian5.img.manifest.xml
```

■ Kernel-Image registrieren:

```
# ec2-bundle-image -i /boot/vmlinuz-2.6.26 --kernel true  
# ec2-upload-bundle -b kernel26 -m /tmp/vmlinuz-2.6.26.manifest.xml  
# ec2-register kernel26/vmlinuz-2.6.26.manifest.xml
```

■ Ramdisk-Image registrieren:

```
# ec2-bundle-image -i /boot/initrd.img-2.6.26 --ramdisk true  
# ec2-upload-bundle -b ramdisk26 -m /tmp/initrd.img-2.6.26.manifest.xml  
# ec2-register ramdisk26/initrd.img-2.6.26.manifest.xml
```


Registrierte Images prüfen

■ Registrierte Images erhalten einen eindeutigen Bezeichner

- Eucalyptus Machine Image: `emi-xxxxxxxx`
- Eucalyptus Kernel Image: `eki-xxxxxxxx`
- Eucalyptus Ramdisk Image: `eri-xxxxxxxx`

■ Registrierte Images kontrollieren:

```
# ec2-describe-images
```

```
IMAGE    emi-1DE4116D    debian5/debian5.img.manifest.xml
          admin    available    public    x86_64    machine
IMAGE    eki-791612FF    kernel26/vmlinuz-2.6.26.manifest.xml
          admin    available    public    x86_64    kernel
IMAGE    eri-CFBE1450    ramdisk26/initrd.img-2.6.26.manifest.xml
          admin    available    public    x86_64    ramdisk
```

Schlüssel erzeugen und Instanz starten

■ Schlüsselpaar erzeugen und registrieren:

```
# ec2-add-keypair mykey > mykey.private
# chmod 0600 mykey.private
# ec2-describe-keypairs
KEYPAIR mykey
    33:da:6e:13:96:e6:f7:3b:b7:34:a6:28:ba:2f:64:ab:83:70:ef:70
```

■ Instanz starten:

```
# ec2-run-instances emi-1DE4116D --kernel eki-791612FF
                                --ramdisk eri-CFBE1450
                                -k mykey -n 2 -t m1.small
```

- -k <Schlüsselpaar>
- -n <#Instanzen>
- -t <InstanzTyp>
- -z <AvailabilityZone>

Instanzen kontrollieren und nutzen

■ Instanzen kontrollieren:

```
# ec2-describe-instances
```

```
RESERVATION    r-3DDE07D9      admin          default
INSTANCE       i-4901084F      emi-1DE4116D   0.0.0.0        141.52.166.160
running        mykey           0              m1.small
2009-05-13T13:50:37+0000    eki-791612FF   eri-CFBE1450
RESERVATION    r-42FA0732      admin          default
INSTANCE       i-463B08BE      emi-1DE4116D   0.0.0.0        141.52.166.161
running        mykey           0              m1.small
2009-05-13T13:50:10+0000    eki-791612FF   eri-CFBE1450
```

■ Per SSH auf einer Instanz arbeiten:

```
# ssh -i mykey.private 141.52.166.160
```

■ Instanzen beenden:

```
# ec2-terminate-instances i-4901084F i-463B08BE
```

Was tun, wenn's hängt?

■ Log-Dateien von Eucalyptus

`/opt/eucalyptus/var/log/eucalyptus/`

- Cloud-Controller, Cluster-Controller und Node-Controller haben eigene Log-Dateien

■ Log-Dateien von Xen

`/var/log/xen/`

- Erfahrungen mit den Eigenheiten von Xen ist hilfreich

■ Forum von Eucalyptus

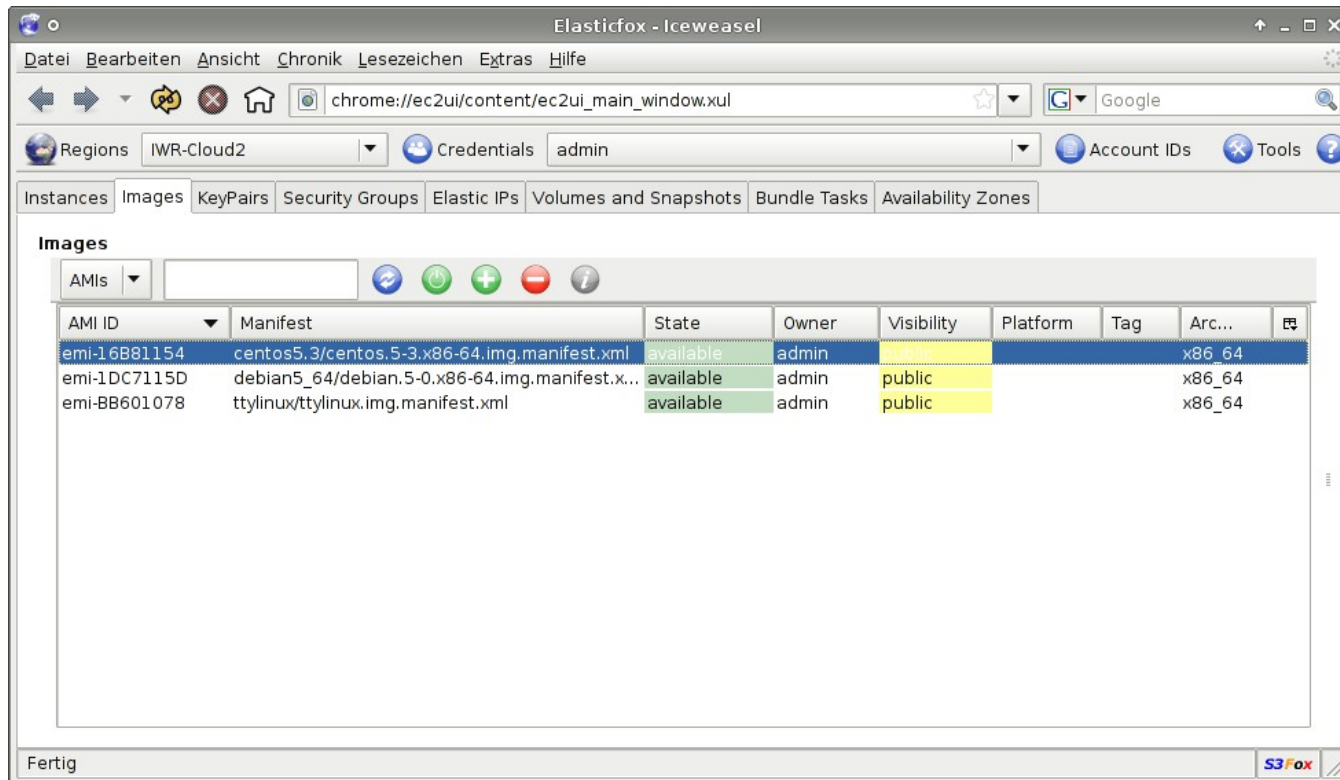
`http://forum.eucalyptus.com/forum/`

■ Kaffee, Zeit und gute Nerven...

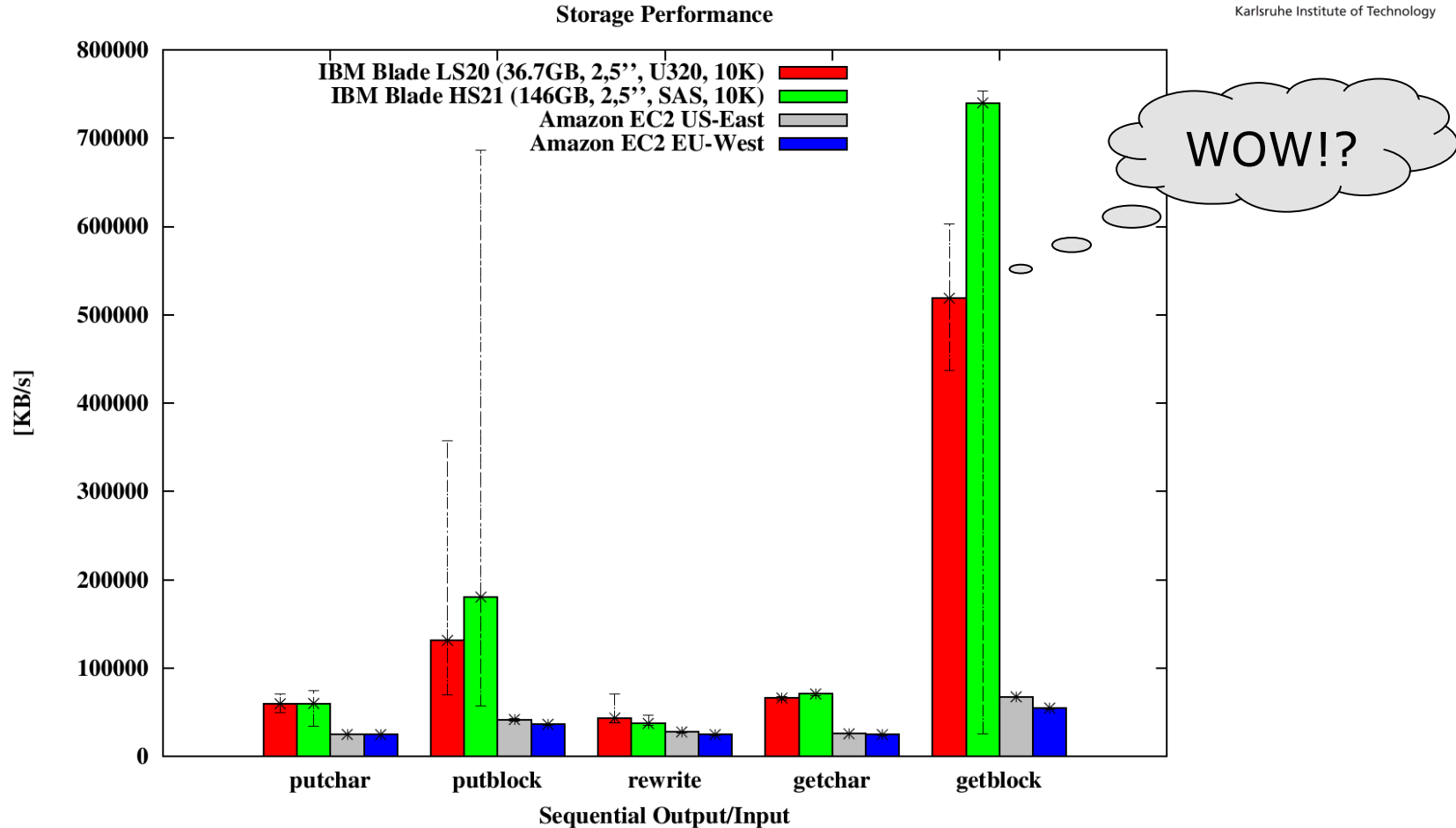
Bekannte Werkzeuge für Eucalyptus

- Die Amazon Web-Services (EC2, S3, EBS, ...) sind äußerst populär
- Im Umfeld der AWS sind viele hilfreiche Werkzeuge entstanden
- Dank der Schnittstellen-Kompatibilität von Eucalyptus zu Amazon EC2, S3 und EBS können viele dieser Werkzeuge auch mit Eucalyptus eingesetzt werden
 - Warum nicht alle? Wegen der unterschiedlichen API-Versionen
- Hilfreiche Werkzeuge:
 - ElasticFox (Firefox-Plugin)
 - <https://code.launchpad.net/~soren/elasticfox/elasticfox.eucalyptus>
 - S3 Curl (Kommandozeilen-Werkzeug)
 - <http://developer.amazonwebservices.com/connect/entry.jspa?externalID=128>
 - s3cmd (Kommandozeilen-Werkzeug)
 - <http://s3tools.org/s3cmd>
 - s3fs (S3-Bucket mit FUSE als lokales Dateisystem mounten)
 - <http://code.google.com/p/s3fs/>

- Erleichtert das Starten, Beenden und Überwachen von Instanzen
- Zusammenarbeit mit Eucalyptus noch etwas unausgegoren
- Version 55 arbeitet mit Eucalyptus 1.5.1 recht gut zusammen
- Mit Eucalyptus 1.6 (September 2009) soll alles besser werden
 - Kompatibilität zur Amazon AWS Spezifikation 1/1/2009



Speicher-Performance S3 vs. Eucalyptus



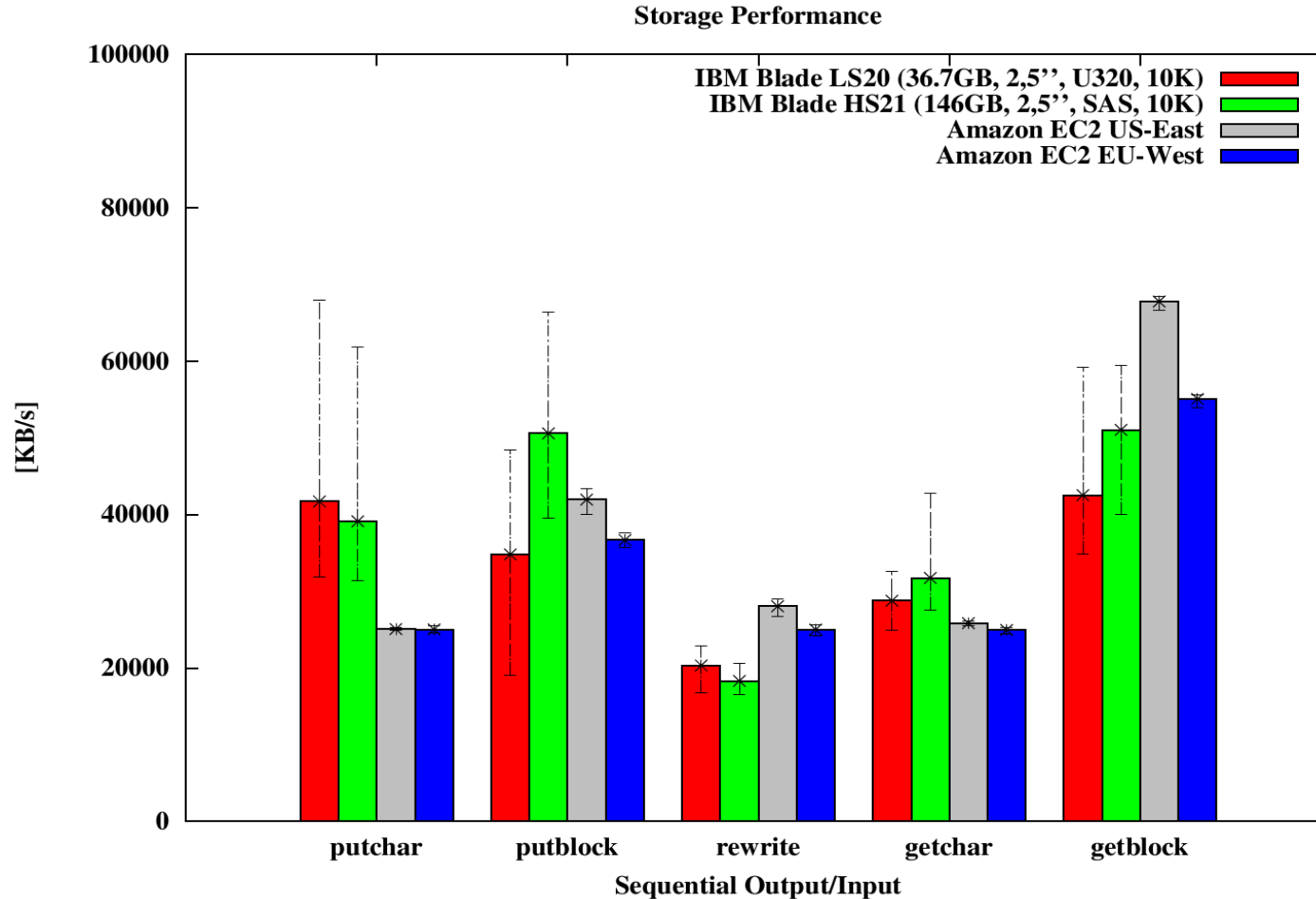
Sequentielle Ausgabe

- Zeichenweise: Datei mit `putc()` schreiben
- Blockweise: Datei mit `write()` schreiben
- Rewrite: `read()` und `write()`

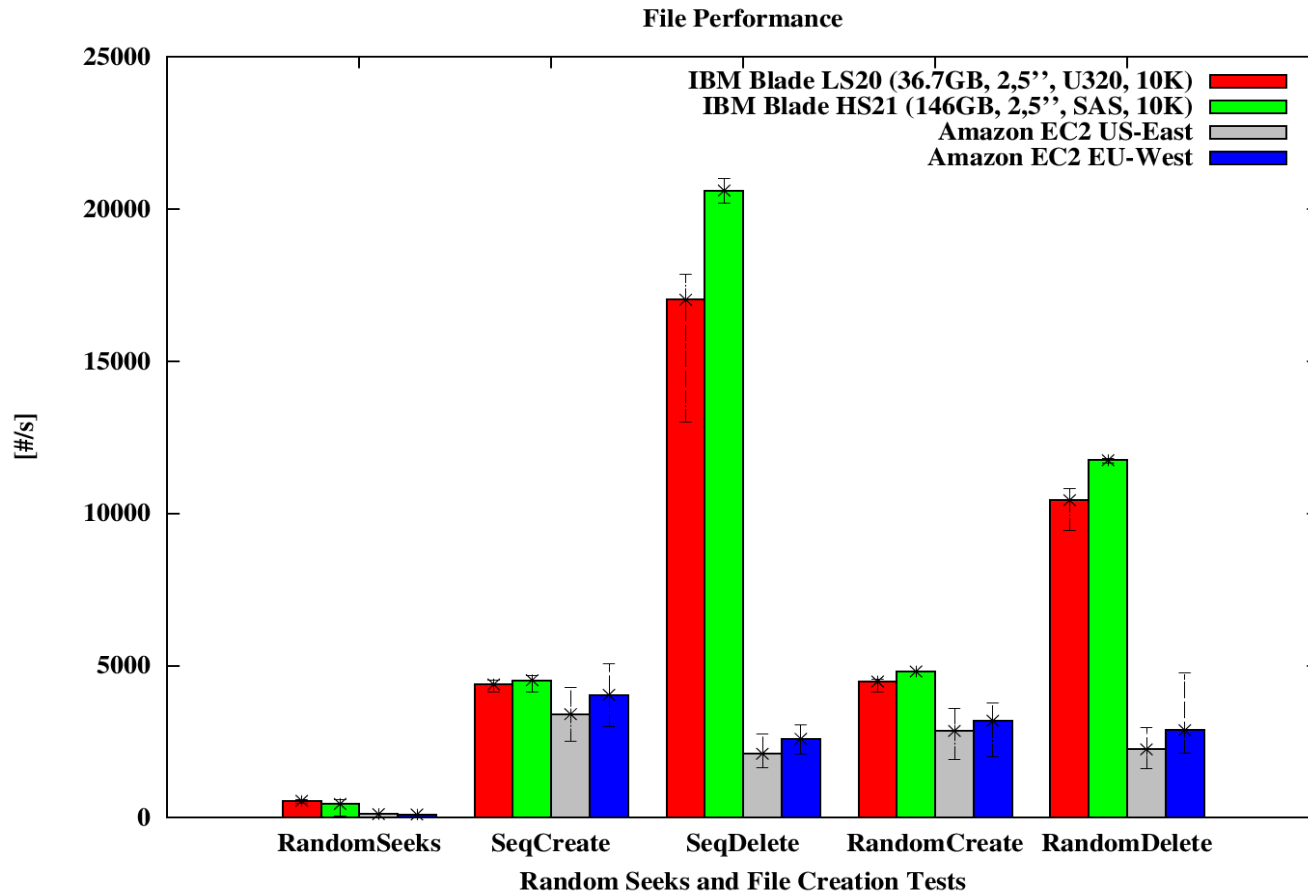
Sequentielle Eingabe

- Zeichenweise: Datei mit `getc()` lesen
- Blockweise: Datei mit `read()` lesen

Realistische Werte...

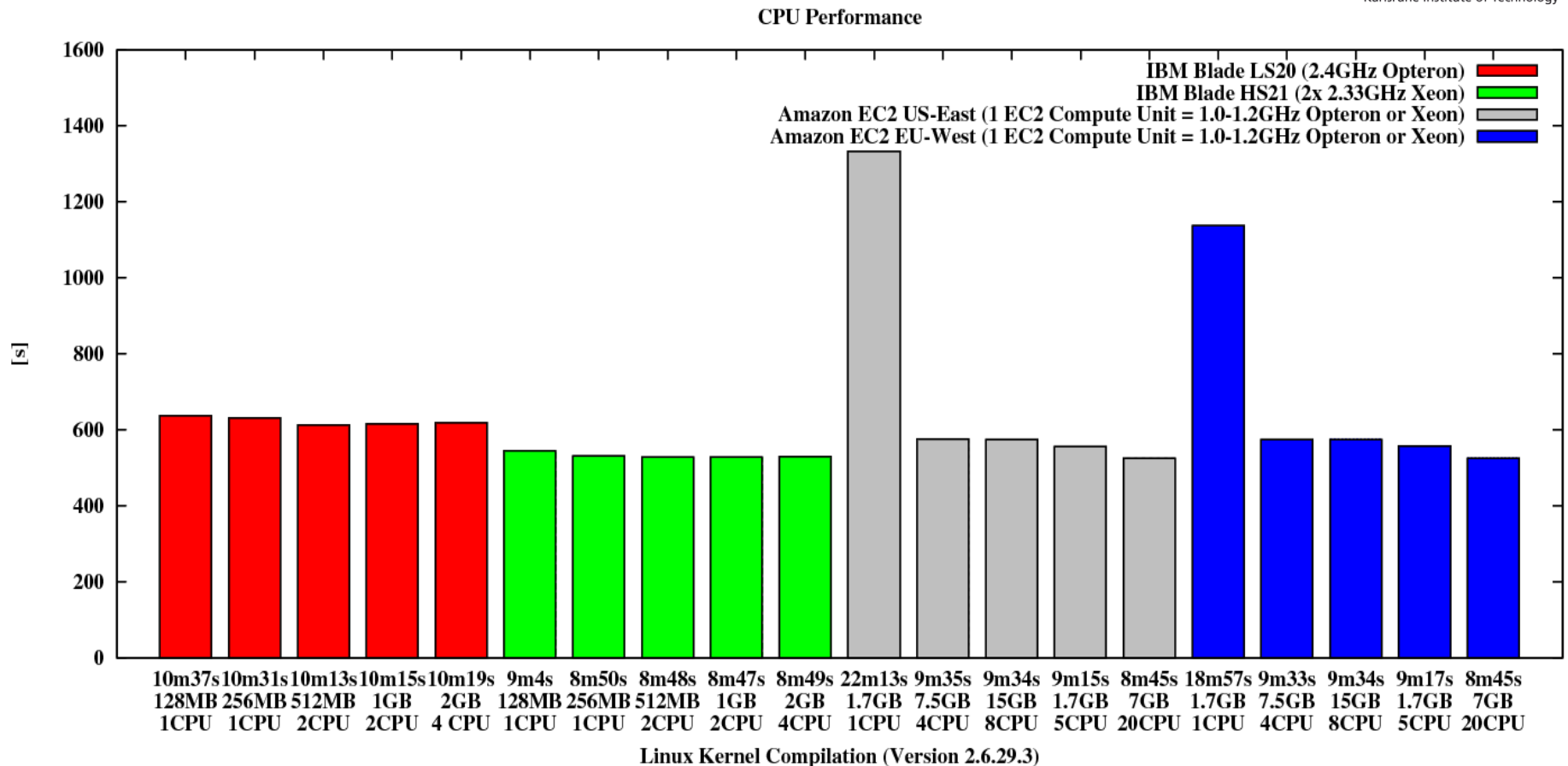


- Der Hauptspeicher des Eucalyptus NCs wurde reduziert um das Caching durch den Kernel zu überwinden
- Die Performance hängt vom eingesetzten Speicher-Subsystem ab
- Die Schreibgeschwindigkeit in Eucalyptus ist in diesem Szenario etwas besser



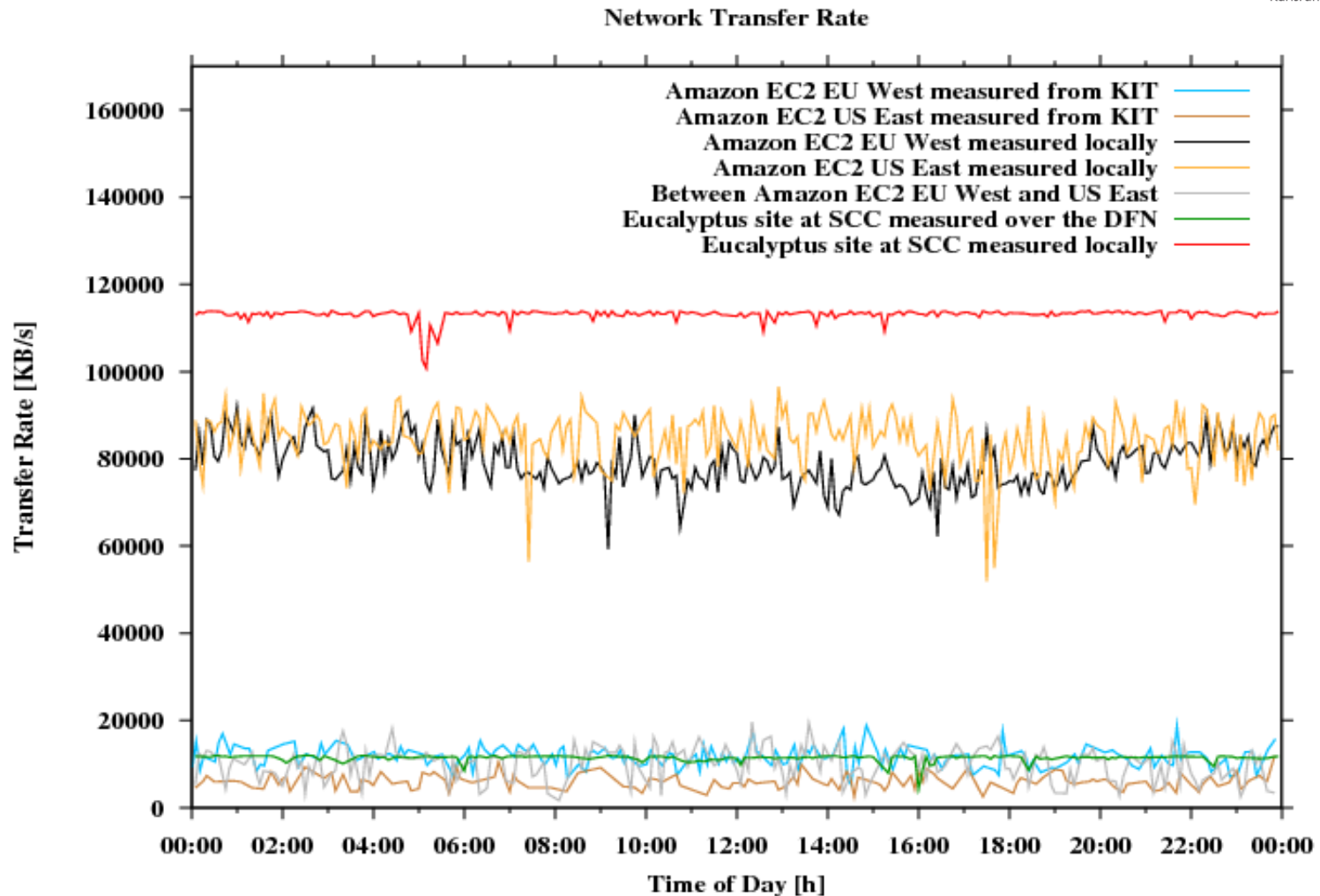
- Eucalyptus bietet bei wahlfreiem Zugriff (Random Seeks) und Datei-Operationen (Dateien erzeugen/löschen) eine bessere Performance als S3
- Möglicherweise wegen der physischen Nähe von EC2 und S3
- Leistungsfähigkeit und Auslastung von Amazon S3 ist uns nicht bekannt

CPU Performance



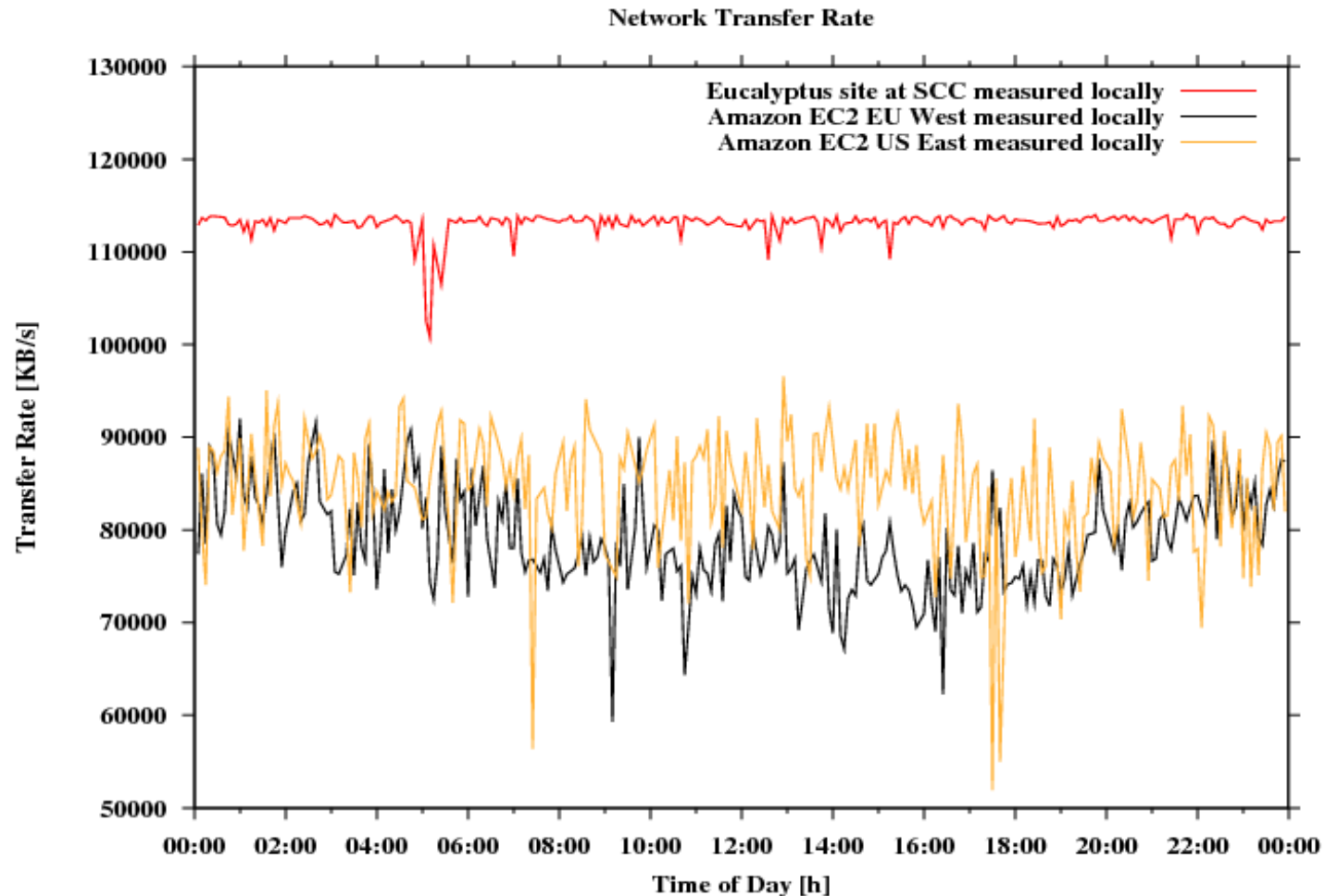
- Performance der Eucalyptus-Infrastruktur (Instanz-Klassen) ist für diesen Aufbau (siehe CPUs) vergleichbar mit Amazon EC2
- Interessant: Mehr CPUs und Hauptspeicher führen nicht zwangsläufig zu einem großen Wachstum der Performance

Durchsatzrate (Network Transfer Rate)



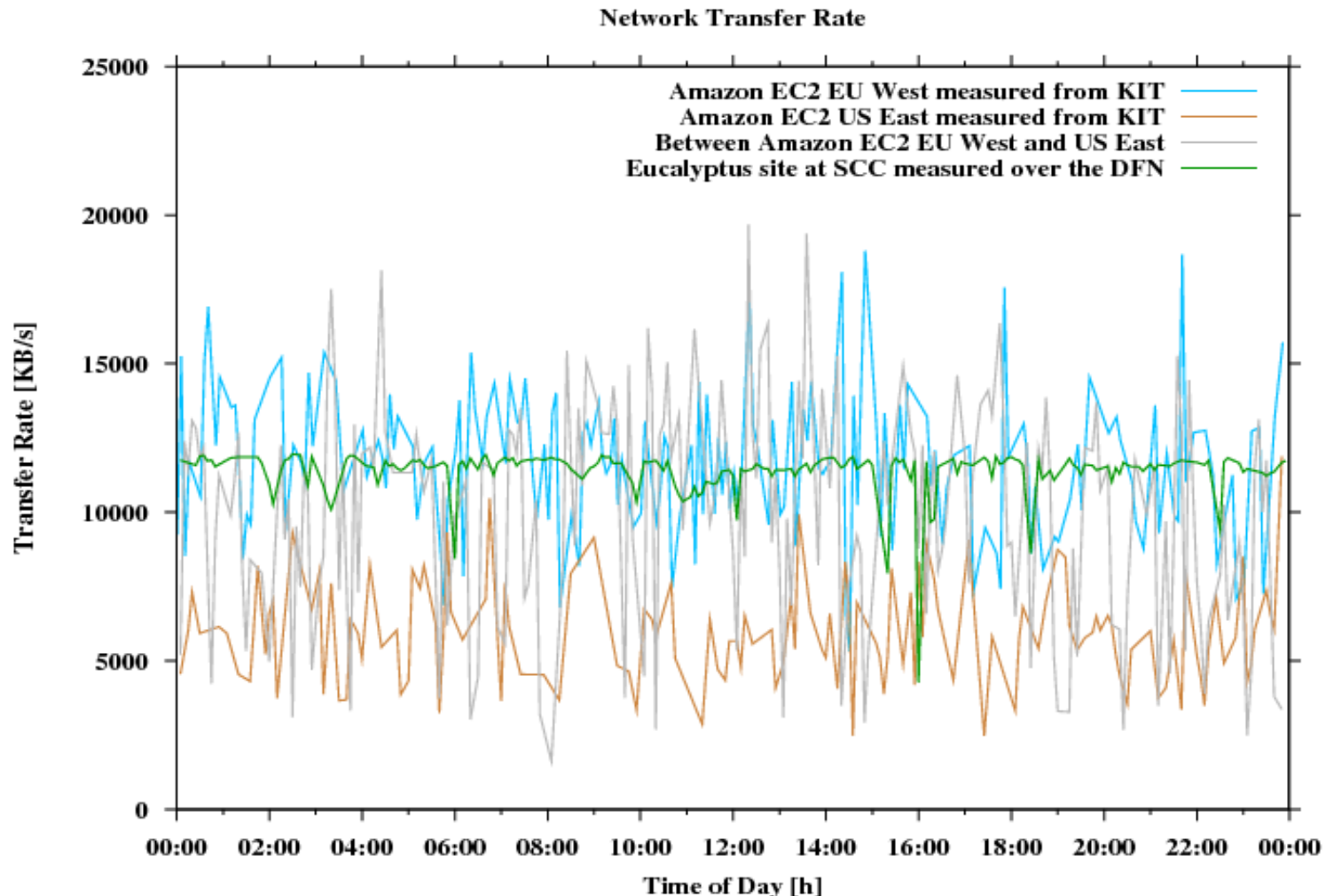
- Durchsatzrate von/zu Eucalyptus im FZK und Amazon EC2 (auch innerhalb)
- Etwas sehr unübersichtlich...

Durchsatzrate im Detail (1)

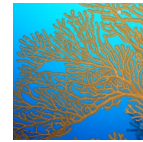


- Durchsatzrate innerhalb Eucalyptus ist konstanter als bei Amazon EC2
- Grund für die hohe Durchsatzrate im Eucalyptus-Cluster: 1000 Mbit/s
- In den EC2-Standorten vermutlich auch 1000 Mbit/s, aber höhere Auslastung

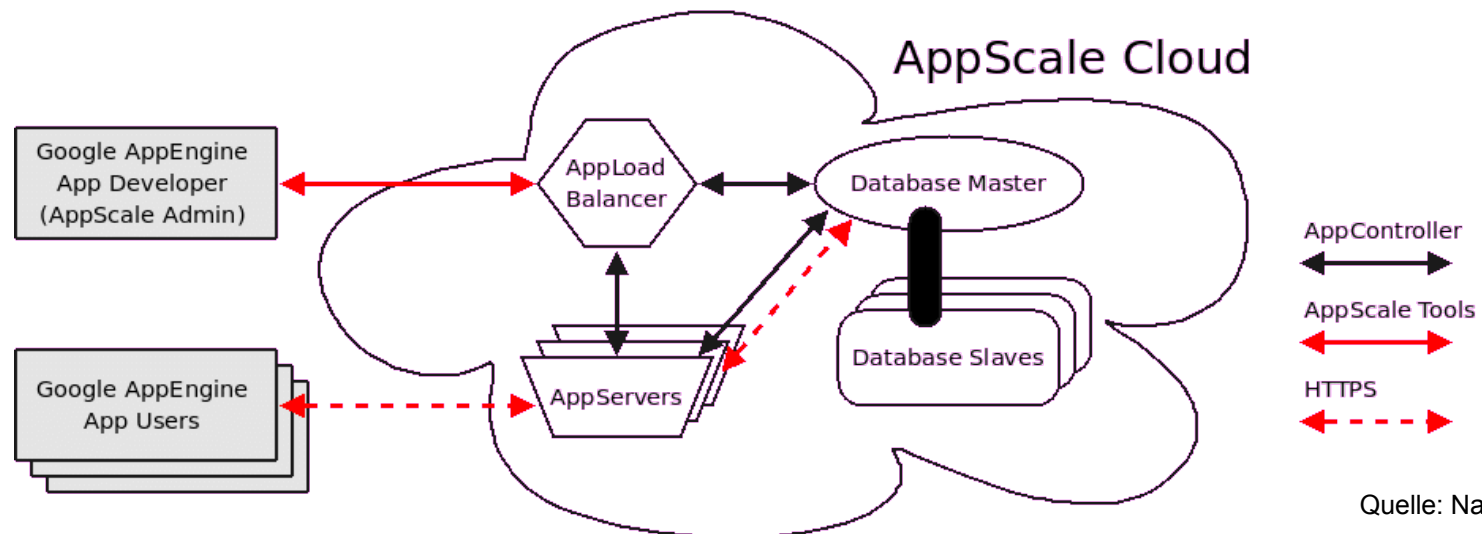
Durchsatzrate im Detail (2)



- Das EC2 EU ist von Europa aus schneller zu erreichen als EC2 US (logisch)
- Durchsatzrate zu Eucalyptus über DFN ist konstanter als zu Amazon EC2



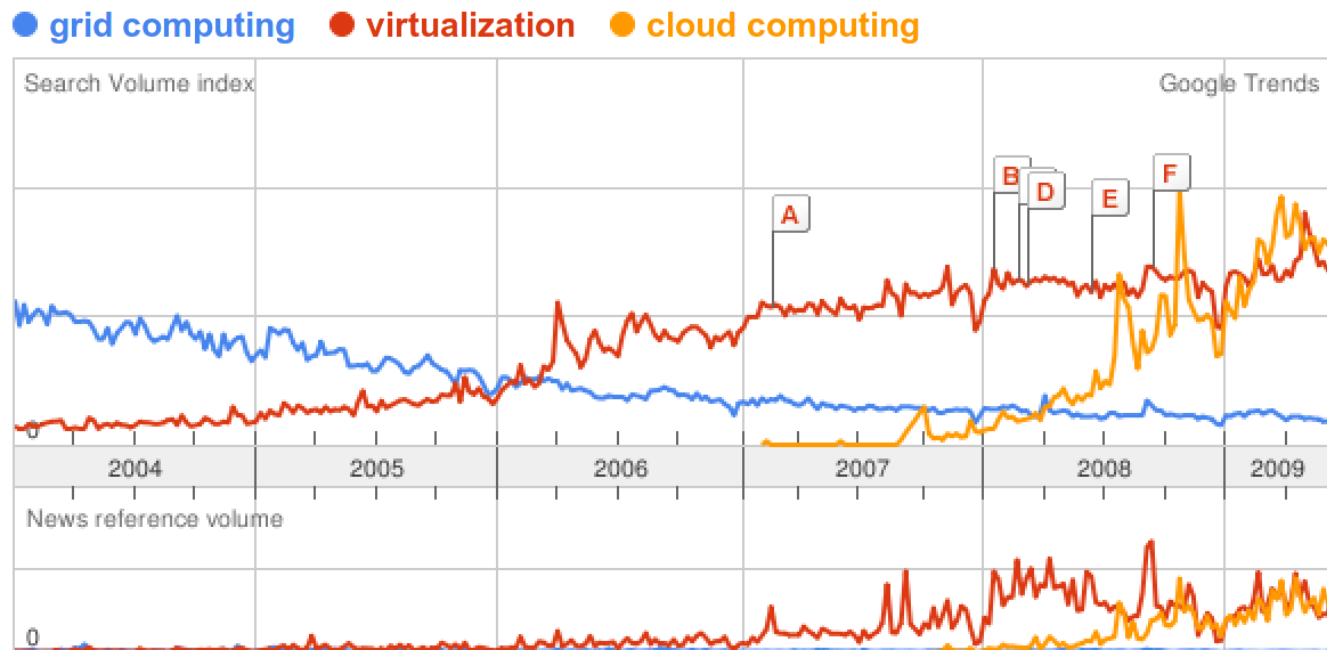
- Open-Source Reimplementierung von Google App Engine
- Entwickelt an der UC Santa Barbara
- App Engine ermöglicht es, Web Anwendungen in Python (und JAVA) auf der Google Infrastruktur zu betreiben
- AppScale arbeitet transparent in Cloud-Infrastrukturen wie Eucalyptus
- AppScale ermöglicht den Aufbau einer eigenen Platform-as-a-Service (PaaS) Cloud-Infrastruktur, die kompatibel zu Google App Engine ist
- Anwendungen für Google App Engine können innerhalb einer Private Cloud betrieben und getestet werden



Quelle: Navraj Chohan

Fazit

- Cloud-Computing ist aktuell ein großes Thema
- Vielversprechender Ansatz, einige drängende Probleme der IT zu lösen
- Cloud-Computing kann helfen, die IT zu industrialisieren
- Eucalyptus und AppScale ermöglichen IaaS und PaaS als OpenSource-Lösungen unter Linux
- Aber: Es gibt noch viele Baustellen im Cloud-Bereich



Vielen Dank für Ihre Aufmerksamkeit

