Infrastructure as a Service (IaaS)

Am Beispiel von Amazon EC2 und Eucalyptus

Jörg Brendel
Joerg.Brendel@informatik.stud.uni-erlangen.de

ABSTRACT

Infrastructure as a Service (IaaS) ist eines der Schlüsselkonzepte im Kontext von Cloud Computing und bietet die Möglichkeit Computing Clouds zu klassifizieren.

Es ist Bestandteil des Konzeptes *Everything as a Service* (XaaS), dass die gesamte Struktur einer Cloud prägt. Um zu zeigen, wie eng dieses Konzept mit Cloud Computing verknüpft ist, werden Amazon EC2 und Eucalyptus im Hinblick darauf betrachtet. Darüber hinaus wird ein Vergleich zur traditionellen, komponentenorientierten Infrastruktur gezogen und es wird erörtert, ob die bisherige komponentenorientierte Infrastruktur vollständig auf eine virtuelle Infrastruktur in der Cloud abbildbar ist.

Im weiteren wird gezeigt, bei welchen Komponenten durch Virtualisierung eine Veränderungen des Systemverhaltens zu erwarten ist und welche weiteren Einflüsse die konsequente Umsetzung von Everything as a Service hat. Hierfür werden die Abrechnungsmodelle traditioneller, komponentenorientierte Infrastruktur mit Angeboten virtueller Lösungen und dem Angebot von Amazon EC2 verglichen.

1. EINFÜHRUNG

IT-Outsourcing an Hosting-Provider (Provider) ist ein seit vielen Jahren etabliertes Geschäftsmodell, dass sowohl im regionalen Markt mit kleineren lokalen Anbietern, wie auch im internationalen Markt mit großen Anbietern existiert. Provider bieten eine Vielfalt von Diensten an. Das Angebot reicht von der Bereitstellung von einfachen Basisdiensten wie zum Beispiel Webspace und E-Mail über Application-Hosting (*Software as a Service*, SaaS) und Datenreplikation (*Content Delivery Networks*, CDN) bis hin zu Servern und komplexer Infrastruktur, bestehend aus Switches, Routern und Loadbalancern.

Das Angebot teilt sich in zwei Kategorien: Zur ersten Kategorie gehören Angebote die stark softwareorientiert sind und in Form von anteiligen Ressourcen eines Servers zur Verfügung gestellt werden. Ein Beispiel dafür sind E-Mail-Dienste und Webspace. Zur zweiten Kategorie gehören Angebote für dedizierte Komponenten, wie zum Beispiel Server.

Der wesentliche Unterschied zwischen den angebotenen Diensten liegt dabei in der Art, wie diese bereitgestellt werden. Softwareorientierte Dienste sind beliebig oft instanzierbar, so dass sich die Einrichtung einer E-Mail oder das Anlegen eines Webspaces auf wenige Einträge in einer Konfigurationsdatei oder Datenbank reduziert, während das zur Verfügung stellen eines Servers mit der Beschaffung und Bereitstellung einer physikalischen Komponente durch einen Techniker im Rechenzentrum des Providers verbunden ist.

Die Instanziierung eines Dienstes ist bei Bedarf in sehr kurzer Zeit möglich und erfüllt somit das *Bei Bedarf*-Paradigma. Die Bereitstellung eines Servers hingegen ist im Allgemeinen nur mit einer gewissen Verzögerung möglich. Weitere Unterschiede finden sich in den Mietmodellen. Softwareorientierte Dienste haben kurze Mindestmietperioden und geringe oder keine Einrichtungskosten, während ein Kunde im Komponentenbereich im Allgemeinen langfristige Mietverträge mit Laufzeiten von einem bis mehreren Jahren und wesentlich höheren Einrichtungskosten hinnehmen muss.

Anbieter von Cloud Computing versprechen jetzt eine Lösung für diese Diskrepanz: Virtualisierung ermöglicht es, einen Server wie eine Software zu instanziieren. Aus der physikalischen Infrastruktur (PI) wird hierdurch eine virtuelle Infrastruktur (VI). Im Rahmen von Cloud Computing betritt ein neues Konzept mit virtueller Infrastruktur als Dienstleistung (Infrastruktur as a Service, IaaS) den Markt und bietet dem Kunden fortan die Möglichkeit, einen Server wie Software zu behandeln. Dieses Konzept findet sich auch in den Abrechnungsmodellen wieder: Infrastruktur as a Service wird mit kurzen Zeitmietmodellen und in der Regel ohne Einrichtungsgebühren angeboten.

Aber virtuelle Infrastruktur wirft auch neue Fragen auf. Physikalische Infrastruktur kann hierarchisch organisiert werden: Loadbalancer stellen die Verbindung nach außen her und verteilen auf verschiedene Application-Server, Kunden mit eigenem IP-Adressbereich (*Autonomes Subnetz*, AS) benötigen eigene Router, VPN-Router ermöglichen einen geschützten Zugang, Firewall-Appliances schützen die hinter ihnen liegende Infrastruktur.

Virtuelle Infrastruktur hat eine andere Topologie und es handelt sich um Ressourcen-Sharing. Daraus ergibt sich die Frage, ob es möglich ist, physikalische Infrastruktur vollständig auf eine virtuellen Infrastruktur abzubilden und welche Veränderungen bezüglich des Laufzeitverhaltens durch die Virtualisierung zu erwarten sind.

2. EVERYTHING AS A SERVICE

Hinter Everything as a Service (XaaS) [16] verbirgt sich die Idee, alles als im Rahmen eines Dienstleistungskonzepts zur Verfügung zu stellen. Die Dienstleistung oder Ressource, die zur Verfügung gestellt werden soll, wird von einem Provider bevorratet und bei Bedarf an einen Kunden vermietet. Im Allgemeinen sind dafür kurze bis sehr kurze Mindestmietperioden vorgesehen und die Leistung kann *jederzeit* bereitgestellt werden. Die Konfiguration der Dienstleistung geschieht über Service-Webseiten durch den Kunden (*Kundenselbstbedienung*, Self-Service [19]) Für den Kunden der

ein solches Angebot nutzt bedeutet dies *keine Kapitalbindung*, da die Finanzierung und Bevorratung als Dienstleistung durch einen Provider erbracht wird und zugleich eine *Risikoreduzierung*, da keine langfristigen Verträge und daraus resultierende finanzielle Verpflichtungen eingegangen werden.

2.1 Beispiele für XaaS

Im Folgenden werden einige Dienste, die die Bedingungen für Everything as a Service erfüllen, betrachtet. Wichtig für einen Dienst, ist dass die Kriterien *bei Bedarf*, *Self-Service* und *Risikoreduzierung* erfüllt werden. Wenn ein Provider seine Dienste unter diesem Aspekt anbietet, ist auch Sicherzustellen, dass die Leistung zeitnah, auch bei hohem Bestellvolumen, erbracht werden kann. Der Dienst muss zusätzlich unter dem Kriterium Dienstgüte und Verfügbarkeit bewertet werden, dazu definieren Provider für einen Dienst eine *Dienstgütevereinbarung* (engl. Service Level Agreement, SLA), als Maß der Verfügbarkeit und der zu erwartenden Leistung.

2.1.1 Basic Services

Schon seit Mitte der 90er Jahre werden Dienste im Rahmen von Ressource-Sharing angeboten. Dies waren vor allem einfache Webspace- und E-Maildienste. Diese werden heute durch die verfügbaren hohen Bandbreiten um Dienste wie zum Beispiel Online-Speicherplatz ergänzt.

Solche Dienste können in der Regel online über eine Service-Webseite eines Providers bestellt werden und stehen danach unmittelbar zur Verfügung. Der Kunde kann diese Dienste bei Bedarf aktivieren, und so auf seine aktuelle Bedarfssituation reagieren und genau die Ressourcen mieten, die er aktuell benötigt.

Durch die kurze Mindestmietperiode können nicht mehr benötigte Ressourcen zeitnah wieder freigegeben werden. Möglich wird dies durch einen hohen Automatisierungsgrad verbunden mit Kundenselbstbedienung.

2.1.2 Humans as a Service

Menschliche Arbeitskraft wird überall dort eingesetzt, wo nicht alle Arbeitsschritte automatisierbar sind oder der Mensch die Arbeit effizienter ausführen kann als der Computer.

Die menschliche Arbeitskraft muss aber nicht sichtbar sein. Ein gutes Beispiel hierfür ist der Administrator, der initial einen Rechner in einem Rechenzentrum in Betrieb nimmt, bevor die weitere Konfiguration automatisiert erfolgen kann. Dies wird als *Humans as a Service* (HuaaS) verstanden.

2.1.3 Software as a Service

Software ist beliebig oft kopierbar, ohne dass erneut Kosten für Rohstoffe oder Produktion anfallen. Dadurch ist Software ideal dafür geeignet, um in einem Mietmodell vertrieben zu werden. Fast alle großen Softwarehersteller bieten ihre Produkte auch auf diesem Weg an. Diese Vertriebsart wird als Software as a Service (SaaS) bezeichnet. Man kann aber auch die auf einem gemieteten Server vorinstallierte Ausführungsumgebung als Software as a Service verstehen.

2.1.4 Hosting Infrastructure as a Service

Die gesamte Infrastruktur eines Provider, die benötigt wird, um Server zuverlässig zu betreiben und an das Internet anzubinden, kann man als eine *As-a-Service*-Dienstleistung verstehen. Beispiele dafür sind Gebäude, Stellflächen, zentrale Notstromversorgung und Klimatisierungsanlagen und ähnliches. Analog zu Humans as a Service ist die Dienstleistung meist nicht direkt sichtbar, zum Betrieb des eigentlichen Services jedoch zwingend notwendig.

2.2 Service-Stack

Hinter einem Dienstleistungsangebot verbergen sich häufig mehrere Services. Damit das Angebot den Kriterien von Everything as a Service genügt, müssen alle nach außen sichtbaren Services ebenfalls diese Kriterien erfüllen. Insgesamt bilden die Services eine Hierarchie und werden als *Service-Stack* [17] betrachtet. Der Service-Stack ist von außen für den Benutzer unsichtbar. Das Dienstleistungsangebot wird über eine Webseite durch den Kunden konfiguriert, die konkrete Umsetzung und Konfiguration der verschiedenen Service-Layer im Service-Stack erfolgt für Benutzer somit transparent.

3. INFRASTRUCTURE

Für die Erklärung des Begriffs *Infrastructure* liefert ein englisches Wörterbuch die folgende Umschreibung: „the basic physical and organizational structures (e.g. buildings, roads, power supplies) needed for the operation of a society or enterprise “[10]. Im IT-Umfeld wird der Begriff *IT-Infrastruktur* [18] verwendet. Dieser definiert alle zum Betrieb der Anwendungssoftware nötigen Güter als *IT-Infrastruktur*.

3.1 Physikalische Infrastruktur

Um den Begriff IT-Infrastruktur enger zu fassen, wird der Begriff Physikalische Infrastruktur (PI) definiert. Unter der physikalischen Infrastruktur versteht man einen oder mehrere Server und alle zur Kommunikation mit anderen Rechensystemen oder dem Benutzer notwendigen Komponenten. Neben den Servern werden folgende Komponenten verwendet: Switches, Router, Firewalls, Proxies, Loadbalancer, Speichernetze (SAN) und passive Komponenten (Kabel, Patchfelder)

Diese Komponenten zeichnen sich dadurch aus, dass sie in der Regel für eine dedizierte Aufgabe gedacht sind. Entsprechend ist die Konfiguration nur für einen Administrator-Zugang vorgesehen, es gibt keine Ebenen bei der Administration. Ressource-Sharing ist nur für die gesamte konfigurierte Komponente möglich. Es gibt keine Möglichkeit zur partiellen Individualisierung durch den Benutzer.

Eine besondere Rolle spielen in diesem Kontext die passiven Komponenten, durch die die aktiven Komponenten hierarchisch verschaltet werden. Dieser Schritt erfolgt manuell und ist nicht bei Bedarf im Sinn von Konfiguration einer Software oder einer Komponente änderbar, sondern erfordert einen manuellen Eingriff direkt am Standort der jeweiligen Komponente. Dafür hat man jedoch einen hohen Freiheitsgrad und kann jede gewünschte Topologie umsetzen.

3.1.1 Abrechnungsmodell für physikalische Infrastruktur

Bezüglich der Hosting-Angebote mit physikalischer Infrastruktur existieren grundsätzliche zwei Abrechnungsmodelle. Wird physikalische Infrastruktur anteilig genutzt, so wird sie in der Regel vermietet. Alle Benutzer der physikalischen Infrastruktur finanzieren dabei in Form ihrer Miete die physikalischen Infrastruktur inklusive der Betriebskosten und der Instandhaltung. Solche Kosten werden bei Inanspruchnahme meist in Form von Grundgebühren abgerechnet. Wird physikalische Infrastruktur exklusiv benutzt, dann stellt der Provider entweder nur die Stellflächen zur Verfügung (*Colocation*) und der Kunde finanziert seine Komponenten selbst oder der Provider übernimmt die Finanzierung und stellt physikalischer Infrastruktur zur Miete oder zum Mietkauf zur Verfügung. In den Kosten für diese Variante sind meistens auch die zuvor genannten Kosten enthalten. Grundsätzlich haben Verträge für physikalische Infrastruktur, die durch den Provider finanziert wird relativ lange Mindestmietperioden, die im Bereich von Monaten bis hin zu mehreren Jahren liegen. Dabei gilt: je exklusiver die Komponente benutzt wird und je teurer die Komponente ist, desto länger sind die Mindestmietperioden.

3.1.2 Physikalische Infrastruktur as a Service

Provider bieten seit vielen Jahren physikalische Infrastruktur in einem Mietmodell an und übernehmen die Finanzierung, die Installation und die Instandhaltung als Dienstleistung. Aber diese Dienstleistung passt nicht zu der am Anfang des zweiten Kapitels gemachten Definition von Everything as a Service: Es werden alle drei wichtigen Kriterien *bei Bedarf*, *Self-Service* und *Risikoreduzierung* verletzt. Die Bereitstellung von physikalischen Infrastruktur dauert länger, da eine Komponenten gekauft und installiert werden muss, ist also nicht bei Bedarf möglich. Die Finanzierung wird im allgemeinen vom Provider übernommen, jedoch sind die Mindestmietperioden deutlich länger, so dass der Kunde eben keine Risikoreduzierung hat. Der Self-Service ist stark eingeschränkt, da bei vielen Komponenten keine Administrationsebenen existieren. Dafür existieren jedoch mehr Freiheitsgrade in der Topologie, auch wenn die Umsetzung durch den Provider erfolgen muss, was mit Verzögerung und Kosten verbunden ist.

3.2 Virtuelle Infrastruktur

Virtualisierung auf Standard-Hardware [20] ermöglicht es, virtuelle Server (VM) wie Software auf einem physikalischen Server (VM-Node) zu instanziieren. Dadurch ist es möglich geworden viele Komponenten der physikalischen Infrastruktur als Softwarelösung in einer virtuellen Maschine zu realisieren. Aus der physikalischen Infrastruktur wird dadurch virtuelle Infrastruktur.

3.2.1 Abbildung physikalischer auf virtuelle Infrastruktur

Virtuelle Maschinen bilden die Basis für alle weiteren Komponenten der virtuellen Infrastruktur, und es soll erreicht werden, dass die selben Komponenten bereit gestellt werden können, die auch in einer physikalischen Infrastruktur vorhanden sind [Tabelle: 1].

PI-Komponente	VI-Komponente
Switch	-
Router	Vyatta [14] , OpenWRT [12]
Firewall	Astaro Virtual Appliance [2]
Proxies	Squid [13]
Loadbalancer	Linux Virtual Server [7]
Speichernetze (SAN)	iSCSI
Passive Komponenten (Kabel, Patchfelder)	-

Tabelle 1: Abbildung physikalische Infrastruktur (PI) auf virtuelle Infrastruktur (VI)

Die Möglichkeit der Abbildung von Switches und passiven Komponenten auf die virtuellen Infrastruktur ist nicht eindeutig zu beantworten. Probleme entstehen durch die hohen Schaltgeschwindigkeiten von einem Switch. Dies ist in Software nicht realisierbar. Das logische Verhalten eines Switches ist abbildbar, z.B. das Bridge-Device unter Linux [3]. Bei der durch entsprechende Verkabelung festgelegten Hierarchie sieht es ähnlich aus. Man kann eine Gruppe von Servern zunächst über einen Switch verschalten und dann die Hierarchie logisch mittels Tagged VLANs [21] erzeugen. Ein solches Netz hat aber ein anderes Laufzeitverhalten und andere Bandbreiten, da hier unter Umständen verschiedene logische Netze auf denselben physikalischen Ports transportiert werden.

3.2.2 Abrechnungsmodell für virtuelle Infrastruktur

Virtuelle Maschinen, realisiert als echte virtuelle Maschine oder Container [11], sind seit einigen Jahren am Markt vertreten. Genau wie andere Komponenten die im Rahmen von Ressource-Sharing zur Verfügung gestellt werden, wird ein Mietmodell mit relativ kurzen Mindestmietperioden angeboten. Die Bestellung und spätere Konfiguration erfolgt über entsprechende Service-Webseiten des Providers. Die Bereitstellung erfolgt im allgemeinen bei Bedarf.

3.2.3 Virtuelle Infrastruktur as a Service

Virtuelle Maschinen erfüllen die Kriterien von Everything as a Service. Ein Provider kann ausreichend physikalische Infrastruktur vorhalten, so dass diese *bei Bedarf* instanziierbar sind. Virtuelle Maschinen werden mit relativ kurzen Mindestmietperioden angeboten und sind meist über eine Service-Webseite für den Kunden selbst administrierbar, so dass auch die Kriterien *Self-Service* und *Risikoreduzierung* erfüllt sind. Eine virtuelle Maschine ist aber nur eine Komponente der virtuellen Infrastruktur, viele Komponenten wiederum können als Softwarelösung in einer virtuellen Maschine realisiert werden. Es ist jedoch noch nicht üblich, dass diese Komponenten in Form von vorkonfigurierten virtuellen Maschinen von einem Provider angeboten werden.

4. INFRASTRUCTURE AS A SERVICE IM KONTEXT VON CLOUD COMPUTING

Cloud Computing kann als Umsetzung von Everything as a Service verstanden werden. Eine Cloud fast verschiedene Servicearten in einem Service-Stack zusammen [Abbildung: 1]. Die verschiedenen Layer des Service-Stacks sind für den Benutzer in der Regel nicht sichtbar, sondern sind hinter

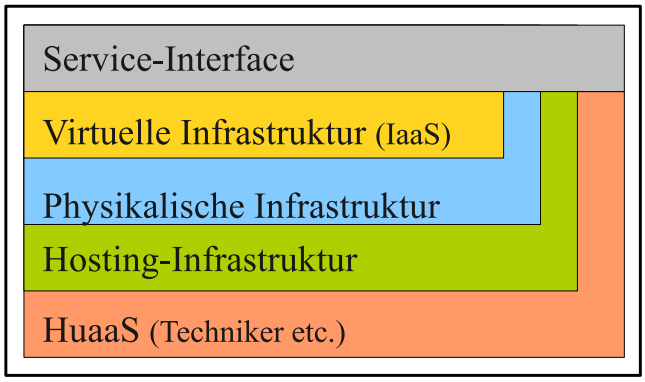

Abbildung 1: Minimaler Service-Stack der Cloud

einer Service-Webseite verborgen.

4.1 Service-Stack in der Cloud

Eine Cloud in der Größe, wie ein Provider sie betreibt, setzt eine skalierbare Infrastruktur voraus. Je nach Ausprägung der Cloud sind unterschiedlich viele Servicearten in der Cloud vorhanden und bilden eine Hierarchie – den Service-Stack der Cloud. Der minimal notwendige Service-Stack umfasst den Infrastructure as a Service-Layer und stellt virtuelle Maschinen bereit. Die virtuelle Maschinen wiederum laufen auf VM-Nodes und benötigen zur Kommunikation und Datenspeicherung weitere physikalische Infrastruktur. Die Cloud hat eine Anbindung an das Internet und wird in einer typische Rechenzentrums-Infrastruktur mit Klimatisierung, entsprechender Stromversorgung, Servicepersonal und Sicherheitseinrichtungen betrieben. Um dem Kunden einen Self-Service zu ermöglichen, wird ein Service-Portal zur Verfügung gestellt, über das der Kunde die gemietete Infrastruktur konfigurieren kann.

4.2 IaaS in der Cloud

Alle Clouds müssen bis zum Infrastructure as a Service-Layer eine sehr ähnliche Service-Struktur implementieren. Dieser Layer muss mindestens eine Ausführungsumgebung für die Applikation, die darauf aufsetzen soll, bereitstellen, kann aber weitere Infrastruktur-Komponenten enthalten. Oberhalb des Infrastructure as a Service-Layers können Clouds verschiedene Ausrichtungen haben. Will man Clouds vergleichen, so kann ab dem Infrastructure as a Service-Layer eine Unterscheidung getroffen werden. Generell findet man hier Angebote zu zwei unterschiedlichen Typen von Clouds: generische Clouds und anwendungsspezifische Clouds

4.2.1 Generische Clouds

Generische Clouds haben die Zielsetzung dem Kunden eine virtuelle Infrastruktur eingebettet in das Everything as a Service-Konzept bereitzustellen. Eine generische Cloud muss einen Service-Stack realisieren, der mindestens alle Layer bis zum Infrastructure as a Service-Layer beinhaltet. Eine ideale generische Cloud ermöglicht es dem Kunden seine physikalische Infrastruktur durch die Infrastruktur der Cloud zu ersetzen, ohne das Änderungen an Komponenten, die oberhalb des Infrastructure as a Service-Layers aufsetzen, nötig sind. Die Abbildung der physikalischen auf die virtuelle

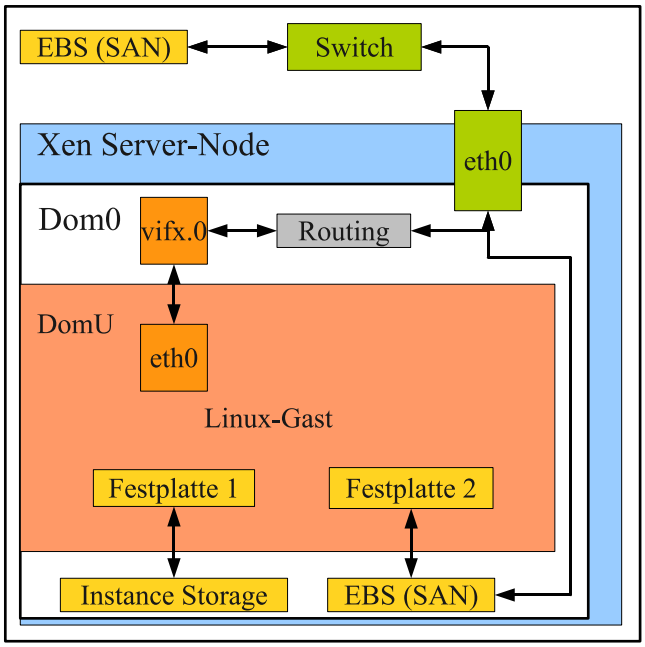

Abbildung 2: Annahme über den EC2 Service-Stack

Infrastruktur muss mindestens hinsichtlich der Belange der Applikation des Kunden vollständig sein. Ein Beispiel für eine generische Cloud ist Amazons EC2 [1].

4.2.2 Anwendungsspezifische Clouds

Eine anderen Ansatz verfolgen anwendungsspezifische Clouds. Der Schwerpunkt liegt hier in der Ausführung einer bestimmten Applikation. Der Service-Stack einer solchen Cloud muss so ausgelegt sein, dass die spezielle Applikation, die als Dienst angeboten werden soll, eine ideale Ausführungsumgebung besitzt. Der Infrastructure as a Service-Layer muss in diesem Fall nur die zum Betrieb der Applikation nötigen Komponenten enthalten und die Ausführungsumgebung kann speziell auf die Applikation optimiert sein. Dieses Modell wird häufig diskutiert, jedoch sind wenige Implementierungen wie zum Beispiel die E-Commerce-Lösung ePages [4] zu finden.

4.3 Amazon EC2

Amazons Elastic Compute Cloud (Amazon EC2) gehört zu der Klasse der generischen Clouds. Entsprechend muss ein Service-Stack, wie zuvor für generische Clouds beschrieben, existieren. Implementierungsdetails zu diesem Thema sind von Amazon nicht offen gelegt, so dass nur Annahmen auf Basis der nach außen sichtbaren Schnittstellen oder der Systemumgebung einer laufenden virtuellen Maschine möglich sind. Amazon veröffentlicht nur die API zur Nutzung der Dienste innerhalb von EC2.

4.3.1 Service-Stack in Amazon EC2

Aus der Systemumgebung einer laufenden virtuellen Maschine kann man erkennen, dass der EC2 zugrunde liegende Hypervisor Xen ist. Aus der Xen-Umgebung lassen sich einige Annahmen über die Speicher- und Netzwerkumgebung machen [Abbildung: 2].

Eine laufende virtuellen Maschine in EC2 hat von innen gesehen eine private IP-Adresse und die Netzwerkanbindung ist in der Terminologie von Xen ein *Routed Network Setup* [15]. Von außen hat die virtuellen Maschine eine öffentliche IP-Adresse. Die Umsetzung muss innerhalb der EC2-Infrastruktur in Form von *Network Adress Translation* (NAT) erfolgen, Details dazu sind für den Benutzer verborgen.

EC2 verfügt über zwei unterschiedliche Arten für die Datenspeicherung: *Instance Storage*, das ist ein lokaler nichtpersistenter Speicher, der nur zur Laufzeit der virtuellen Maschine existiert und *Elastic Block Store* (EBS). Bei dem lokalen Speicher scheint es sich um ein Linux LVM-Volume zu handeln, EBS ist eine Speichernetz-Variante (SAN).

Neben Speicher und virtuellen Maschinen stellt der Service-Stack auch Funktionen zum *Loadbalancing* zur Verfügung. Dabei können mehrere Instanzen zu einem Pool zusammengefasst werden und Netzwerkanfragen werden auf diesen Pool verteilt. Ausgefallene oder gestoppte Instanzen werden automatisch aus dem Pool entfernt. Eine weitere Funktion im Service-Stack ist das sogenannte *Auto Scaling*. Mittels dieser Funktion werden Instanzen je nach Lastsituation bei Bedarf automatisch gestartet oder gestoppt. Im Fall von Loadbalancing werden neuen Instanzen automatisch in den Pool aufgenommen.

4.3.2 Abrechnungsmodell in Amazon EC2

Amazon EC2 unterscheidet 3 Arten von Instanzen: *On Demand Instances*, *Reserved Instances* und *Spot Instances*.

Der typische Fall sind On Demand Instances, die ein Kunde bei Bedarf registrieren kann. Es gibt keine Mindestmietperiode, die Abrechnung erfolgt auf Betriebsstundenbasis. Allerdings gibt es keine Garantie das eine solche Instanz zu jeder beliebigen Zeit aktivierbar ist. Einmal aktiviert laufen sie jedoch weiter bis der Kunde sie stoppt.

Reserved Instances sind eine Form von Reservierung. Der Kunde zahlt eine Gebühr und reserviert dadurch für einen gewissen Zeitraum eine Art Nutzungsrecht. Eine solche Instanz kann während dieses Zeitraums garantiert zu jedem beliebigen Zeitpunkt aktiviert werden. Die Abrechnung erfolgt ab dem Zeitpunkt der Aktivierung auf Betriebsstundenbasis. Gegenüber der On Demand Instances ist der hier berechnete Stundensatz reduziert.

Spot Instances sind eine Idee, um Totzeiten aufzufüllen: Instanzen, die nicht permanent laufen müssen, gestoppt werden dürfen und irgendwann starten können, sind Kandidaten um ungenutzte Rechenkapazitäten auszunutzen. Amazon bietet an, eine Instanz als Spot Instances zu deklarieren und der Kunde kann einen Preis festlegen, den er bereit ist, für die Ausführungsstunde zu bezahlen. Mit sinkender Auslastung reduziert Amazon den Preis und aktiviert die Spot Instances die den aktuellen Preis akzeptieren. Steigt der Preis, werden diese Spot Instances wieder beendet.

4.3.3 Public Cloude

Amazon EC2 ist eine *Public Cloud*. Public Clouds werden für die Benutzer über das Internet zur Verfügung gestellt. Im Fall von Amazons EC2, kann ein beliebiger Benutzer nach Registrierung mit Opt-in E-Mail-Verifikation und nach Hinterlegung von Kreditkartendaten zur Abrechnung, die Dienste der Cloud nutzen. Diese einfache Art der Verifikation ist für den Benutzer angenehm, jedoch ist die Identität des Benutzers nicht sichergestellt. Daraus ergeben sich eine ganze Reihe Implikationen in Bezug auf Sicherheit der Daten in der Cloud:

Die Benutzer sind quasi anonym und Sicherheit einer virtuellen Maschine ist durch die des Hypervisors begrenzt. In der Cloud kann der Benutzer beliebigen Code ausführen. Deshalb können Sicherheitslücken im Hypervisor ausgenutzt werden, um aus dem Hypervisor auszubrechen. Für den Benutzer der Cloud ist nicht ersichtlich ob und welche weiteren virtuellen Maschinen auf dem Server laufen. Eine eigene Sicherheitsbewertung ist für den Kunden nicht möglich, da Versionsstände und Konfigurationsdetails nicht offen gelegt sind. Haftung eines Benutzers ist durch die einfache Verifikation schwer umzusetzen.

4.4 Eucalyptus

Eucalyptus [5] ist ein Projekt der University of California in Santa Babara. Es wurde mit zwei Zielen gestartet: Eucalyptus soll eine Forschungsplattform für Cloud Computing sein und der Open Source Gemeinde ein Cloud Computing-Framework zur Verfügung stellen.

Eucalyptus gehört zu der Klasse der generischen Clouds. Da alle Quellen offen liegen und frei verfügbar sind, ist der gesamte Aufbau der Cloud für den Anwender bekannt. Die nach außen zur Verfügung gestellte API ist zwar zu Amazons EC2 kompatibel, jedoch noch nicht vollständig implementiert.

4.4.1 Service-Stack in Eucalyptus

Eucalyptus implementiert ähnlich wie Amazons EC2 einen Service-Stack bis zum Infrastructure as a Service-Layer. Eucalyptus hat eine modulare Struktur und basiert bezüglich der Virtualisierung auf libvirt [6]. Entsprechend kann die Ausführungsumgebung theoretisch durch jede Ausführungsumgebung realisiert werden, für die ein libvirt-Treiber existiert. Derzeit stehen Xen und KVM [8] zur Auswahl [Abbildung: 3].

Im Netzwerkbereich kennt der Infrastructure as a Service-Layer einen *Virtual Network Overlay*. Realisiert wird dies über ein Bridge-Device und ein VLAN-Interface auf dem Server-Node. Diese Methode hat den Vorteil, dass durch das VLAN-Tagging eine Isolierung der Netze verschiedener VM-Gruppen erreicht wird. Virtuelle Maschinen die zu einer Gruppe gehören, sind dadurch sehr effizient miteinander verbunden. Verbindungen nach außen oder zwischen verschiedenen Gruppen von virtuellen Maschinen werden bei Bedarf als geroutete Verbindung realisiert. Virtuelle Maschinen haben wie in Amazons EC2 private IP-Adressen und die Abbildung auf öffentliche Adressen geschieht mittels NAT-Regeln. Es gibt noch keine Unterstützung für Loadbalancing auf IP-Paket-Ebene wie in Amazons EC2.

Funktionen zur Live-Migration und damit Dienste wie Auto Scaling fehlen ebenfalls. Prinzipiell unterstützt KVM bereits Live-Migration, aber diese Funktionen sind noch nicht über libvirt verfügbar.

4.4.2 Private Cloud

Eucalyptus ist eine *Private Cloud*. Private Clouds werden von Firmen oder Organisationen auf eigener Hardware betrieben. Entsprechend muss eine Private Cloud nicht unbedingt Funktionen zur Abrechnung bereitstellen. Eucalyptus bietet zum Beispiel kein Abrechnungsmodell.

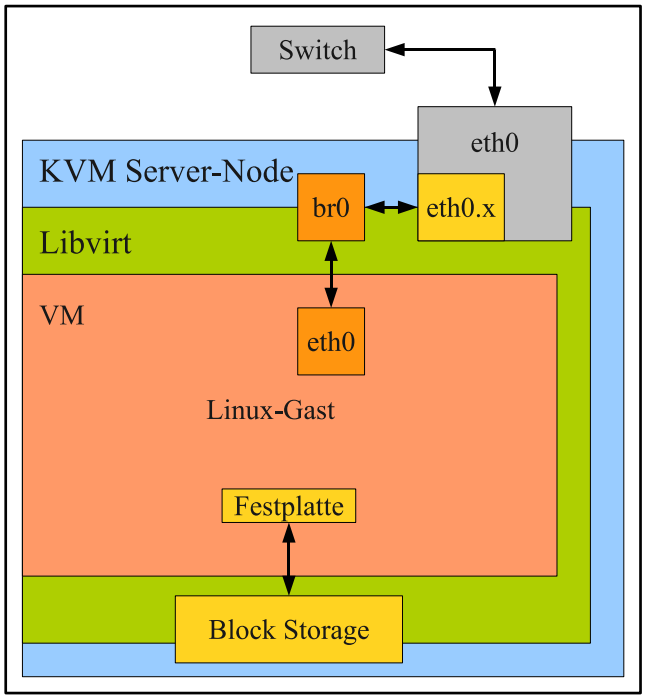

Abbildung 3: Eucalyptus Service-Stack

Viele Gründe sprechen für den Einsatz einer Private Cloud in einer Organisation oder einem Unternehmen: Konsolidierung von Ressourcen, Einsparung von Stellflächen, Strom, Klima und Arbeitszeit für Instandhaltung.

Ein wichtiger Grund für die Zukunft kann die Einführung der Cloud als Technologie sein. Effizientes Nutzen der Cloud, benötigt angepasste Softwarelösungen für verteilte Systeme. Daher kann die Einführung unter dem Aspekt von Wissenserwerb oder der Einführung von neuer Software sinnvoll sein.

Ein weiterer Grund kann *Skalierung bei Bedarf* sein. Es ist denkbar, eine Anwendung bei Durchschnittslast in einer Private Cloud zu betreiben und Lastspitzen mit externen Ressourcen abzudecken. Diese Konzept kann auch unter dem Aspekt Datenschutz Sinn machen. Hierbei bleiben sicherheitskritische Daten in der Private Cloud gespeichert, während die Berechnung auf anonymisierten Daten in der Public Cloud durchgeführt wird. Erweitert man eine Private Cloud um externe Ressourcen einer Public Cloud, so spricht man von einer *Hybrid Cloud*.

5. FAZIT

Cloud Computing ist die Fortführung der Ideen, die bereits die Virtualisierung vorangetrieben haben: Konsolidierung, Einsparung, homogene Ausführungsumgebung (homogen innerhalb der VM) und Ressource-Sharing.

Die wesentliche Neuerung, die durch die Cloud eingeführt wird, ist das Konzept von Everything as a Service. Die am Markt verfügbaren Angebote – besonders im Hinblick auf generische Clouds – haben den Anspruch, den Kunden bei Bedarf Zugriff auf quasi unbegrenzte Ressourcen zur Verfügung zu stellen. Hierfür wird in der Cloud ein Infrastructure as a Service-Layer zur Verfügung gestellt, der eine Ausführungsumgebung in Form einer virtuellen Maschine enthält. Der Kunde ist in der Lage, eine virtuelle Maschinen über eine Service-Webseite des Providers eigenverantwortlich zu verwalten (Self-Service). Ein Argument, was bislang häufig gegen Outsourcing von IT-Infrastruktur sprach, waren Verzögerungen bei Änderungen an der Struktur. Ein Administrator konnte nicht einfach einen weiteren Server in Betrieb nehmen, sondern musste dies bei dem Provider beauftragen und dann warten bis die Ressource zur Verfügung gestellt wurde. Diese Bereitstellungszeit entfällt bei Nutzung der Cloud und damit entfällt auch dieses Argument. Änderungen an der Infrastruktur können also in gleichen oder sogar in kürzen Zeitrahmen realisiert werden, wie bei lokaler Infrastruktur.

Bei der Betrachtung von Amazons EC2 wurde gezeigt, dass der Infrastructure as a Service-Layer keine beliebige Konfiguration im Bereich der Netzwerktopologie erlaubt. Der Layer hat zwar schon erweiterte Funktionalität und bietet Funktionen wie beispielsweise Loadbalancing, ein freies Verschalten des Netzwerks ist aber nicht möglich. Komponenten wie virtuelle Router oder Firewalls existieren bereits am Markt und lassen sich n einer virtuellen Maschine betreiben. Solche Lösungen sind – wenn überhaupt – nur durch den Kunden selber in einer virtuellen Maschine innerhalb der Cloud zu installieren und stellen deshalb keine Komponente des Infrastructure as a Service-Layers der Cloud dar.

Die virtuelle Netzwerkinfrastruktur in Amazons EC2 ist als geroutete Verbindung realisiert und den virtuellen Maschinen werden private IP-Adressen innerhalb der Cloud zugewiesen, die per NAT auf öffentliche IP-Adressen umgesetzt werden. Selbst bei zwei benachbarten virtuellen Maschinen hat man durch die geroutete Verbindung Performance-Einbußen im Netzwerkbereich gegenüber einer direkten Verbindung zu erwarten. Zusätzlich erfolgt das Routing über den Server auf dem die VM läuft, so dass abhängig von der aktuellen Lastsituation mit unterschiedlichen Paket-Laufzeiten zu rechnen ist. Die Umsetzung von der öffentlichen auf die private IP-Adresse mittels NAT kann außerdem zu Problemen bei Multiport-Protokollen führen. Zukünftig besteht daher Bedarf an Virtualisierung von Hardware im Bereich von Switches und Routern, die sich dann als ein Layer in den Service-Stack integrieren lassen.

In Bezug auf Sicherheit innerhalb der Cloud gibt es viele Bedenken. Insgesamt ist die Sicherheit durch die des eingesetzten Hypervisors begrenzt. Zusätzlich müssen mögliche rechtliche Auswirkungen berücksichtigt werden, wenn zum Beispiel personenbezogene Daten an einen Provider auslagert werden, zumal dieser eventuell nicht im gleichen Land ansässig ist.

Generische Clouds sind in der Entwicklung schon sehr weit fortgeschritten und für viele Einsatzszenarien denkbar. Für stark schwankende Lastsituationen bietet sich Cloud Computing als eine gute Lösung an. Insbesondere aus dem Grund, dass Kosten nur bei Bedarf anfallen. Sicherheitsaspekte sind jedoch bei der Speicherung und Verarbeitung personenbezogener Daten zu beachten.

Private Clouds wie Eucalyptus können Cloud-Technologie in die Unternehmen bringen. Beim Einsatz einer lokalen Cloud entfallen die Sicherheitsprobleme der Public Clouds. Eine Verknüpfung von Private und Public Cloud, die Hybrid Cloud, kann eine Lösung für das Problem liefern: Ein Unternehmen kann seine IT-Infrastruktur generell auf Cloud Computing ausrichten und betreibt eine Hybrid Cloud. Kritische Daten werden in der Private Cloud gespeichert und

verarbeitet, zusätzliche Kapazitäten können von einer Public Cloud bezogen werden. In der Public werden nur unkritische und anonymisierte Daten verarbeitet.

Ein Konzept, was noch wenig diskutiert wird, ist die Mischung aus Cloud und physikalischer Infrastruktur. Es existieren viele Managementtools zur automatisierten Einrichtung von physikalischen Rechner, so dass prinzipiell in einer geeigneten Umgebung auch der Einsatz dedizierte Rechner denkbar ist.

Die konsequente Umsetzung von Everything as a Service in Verbindung mit Konzepten wie beispielsweise Auto Scaling in Amazon EC2 und der Möglichkeit von Live-Migration von virtuellen Maschinen schafft neue Möglichkeiten für Energieeinsparungen in Rechenzentren, indem Rechnercluster bei Last dynamisch in Betrieb genommen und bei geringer Last abgeschaltet werden. Da Stromkosten heute maßgeblich die Betriebskosten von Rechenzentren bestimmen, ist Abschalten von ungenutzten Ressourcen der effektivste Weg, Kosten zu sparen. Allerdings führt das Abschalten einer großen Anzahl von Rechner in den heutigen Rechenzentren nur bedingt zum erwünschten Erfolg. Grund dafür ist die Struktur der Rechenzentren. Diese sind dafür geplant worden eine bestimmte Anzahl von Rechner zu betreiben. Der Wirkungsgrad eines Rechenzentrums ist nur für eine bestimmte Belegung optimal und die mittlere Belegung muss möglichst nah an der idealen Belegung liegen. Ein dynamisches Ein- und Ausschalten von Rechnern innerhalb einer kurzer Zeitspanne ist in den heutigen Rechenzentren nicht vorgesehen.

Wenn sich Cloud Computing durchsetzt, wird die Struktur der Rechenzentren gravierend verändert. Ein Rechenzentrum muss zukünftig modular organisiert werden, so dass Energieeinsparungen im vollen Umfang – also unter Beibehaltung eines hohen Wirkungsgrades – bei Abschaltung von unbelegten Ressourcen möglich wird. Ein Beispiel hierfür findet man bereits bei Microsoft. Die neuen Rechenzentren für Microsofts Cloud-Lösung *Windows Azure* sind in Hochseecontainer verpackt [9]. Diese Rechenzentren werden modular aus Funktionsblöcken zusammengestellt: es gibt Container mit Servern, Container mit Stromversorgung und Container mit weiteren Funktionsblöcken.

6. LITERATUR

[1] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/de/ec2. [Online; Stand 01. Mai 2010].

[2] Astaro Security Gateway Virtual Appliance for VMware. http://www.astaro.com/products/astaro-security-gateway-virtual-appliance-for-vmware. [Online; Stand 29. April 2010].

[3] bridge | The Linux Foundation. http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge. [Online; Stand 29. April 2010].

[4] ePages GmbH - e-commerce. now plug and play. http://www.epages.com. [Online; Stand 01. Mai 2010].

[5] Eucalyptus Community. http://open.eucalyptus.com. [Online; Stand 02. Mai 2010].

[6] libvirt: The virtualization API. http://www.libvirt.org. [Online; Stand 02. Mai 2010].

[7] The linux virtual server project — Linux Server Cluster for Loadbalancing. http://www.linuxvirtualserver.org. [Online; Stand 29. April 2010].

[8] Main Page - KVM. http://www.linux-kvm.org. [Online; Stand 02. Mai 2010].

[9] Microsoft eröffnet Rechenzentrum in Chicago - Business | News | ZDNet.de. http://www.zdnet.de/news/wirtschaft_unternehmen_business_microsoft_eroeffnet_rechenzentrum_in_chicago_story-39001020-41509970-1.htm. [Online; Stand 03. Mai 2010].

[10] Online Compact Oxford English Dictionary. http://www.askoxford.com/concise_oed/infrastructure. [Online; Stand 29. April 2010].

[11] OpenVZ Wiki. http://www.openvz.org. [Online; Stand 30. April 2010].

[12] OpenWRT — Wireless Freedom. http://openwrt.org. [Online; Stand 29. April 2010].

[13] Squid: Optimising Web Delivery. http://www.squid-cache.org. [Online; Stand 29. April 2010].

[14] Vyatta.org | The Open Source Networking Community. http://www.vyatta.org. [Online; Stand 29. April 2010].

[15] XenNetworking - Xen Wiki. http://wiki.xensource.com/xenwiki/XenNetworking#head-740e3cf58c2ac48051f74c4f72cc6df52117e87e. [Online; Stand 02. Mai 2010].

[16] Everything as a service — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=Everything_as_a_Service&oldid=71602229, 2010. [Online; Stand 28. April 2010].

[17] Lenk, A. What's inside the cloud? — An Architectural Map of the Cloud Landscape. http://www.hpl.hp.com/personal/Thomas_Sandholm/lenk2009.pdf, 2009. [Online; Stand 28. April 2010].

[18] Patig, S. IT-Infrastruktur. http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/daten-wissen/Informationsmanagement/IT-Infrastruktur. [Online; Stand 29. April 2010].

[19] Wikipedia. Customer self services — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=Customer_Self_Services&oldid=68239925, 2009. [Online; Stand 30. April 2010].

[20] Wikipedia. Commodity computing — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Commodity_computing&oldid=348055943, 2010. [Online; accessed 29-April-2010].

[21] Wikipedia. Virtual local area network — Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/w/index.php?title=Virtual_Local_Area_Network&oldid=72897734, 2010. [Online; Stand 29. April 2010].