

inkl. **CD**

Zweimal Swing: Gestatten SAF und FEST » 84



2.2010

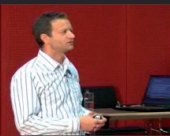
JAVA Mag

JavaTMmagazin

Java • Architekturen • SOA • Agile

www.javamagazin.de

CD-INHALT



Swing App Framework

Session von der
JAX 2009
in voller Länge

DIE HIGHLIGHTS



Citrus
Framework

NoSQL

Alternative Speichersysteme –
Schöne neue Welt? » 14

Jahres-Archiv 2009

Jetzt exklusiv für
Abonnenten online » 2



jax 2010
Alle Infos im Heft

SINN UND UN- SINN VON CLOUD COMPUTING

innoQ

**Sonderdruck
der Firma innoQ**



jCouchDB

Alle CD-Infos ab Seite 3

Security-Modularisierung

Modulares Recycling » 74

Pragmatic SOA

Beschränken auf das Wesentliche » 91

Datenträger enthält
Info- und
Lehrprogramme
gemäß § 14 JuSchG

Schein oder Sein?

Nachdem „SOA“ mittlerweile häufig negative Reaktionen hervorruft, konnte es nicht lange dauern, bis ein frisches Etikett auftaucht, das Anbieter auf ihre Produkte und Dienstleistungen kleben, Konferenzveranstalter als Thema verwenden und Berater in ihr Portfolio aufnehmen können: Cloud Computing. Und auch diesmal gibt es keine akzeptierte Definition sowie viel Neues und viel lange Bekanntes, das es voneinander zu trennen gilt. Grund genug, sich dem Thema aus verschiedenen Richtungen zu nähern.

von Stefan Tilkov

The cardinal economic principle at issue here is that an information utility serving a large number of users can provide service to each more economically than he can provide it for himself, just as a power system can provide energy to its customers at lower cost than they, individually, can generate it for themselves“ – Aus einem Strategiepapier von Western Union, März 1965.

Beginnen wir mit dem Versuch einer Definition: Unter „Cloud Computing“ versteht man allgemein die Nutzung des

Internets – der „Cloud“, nach dem üblicherweise in Diagrammen dafür verwendeten Wolkensymbol –, um Daten abzulegen oder zu verarbeiten. Dabei werden die dafür benötigten Ressourcen dynamisch in der benötigten Menge zur Verfügung gestellt, und die Abrechnung erfolgt auf Basis der tatsächlichen Nutzung.

Irgendetwas Neues hier?

So richtig aufregend klingt das zunächst einmal nicht. Der Unterschied liegt in

der Art und Weise, in der diese Fähigkeiten bereitgestellt werden. Das lässt sich am besten anhand konkreter Beispiele illustrieren.

Amazon AWS: IaaS

Amazon.com bietet mit den Amazon Web Services (AWS) eine Reihe von Diensten an, die sich unmittelbar nach einer Registrierung nutzen lassen. Bekanntestes Beispiel ist die Elastic Computing Cloud (EC2), ein für Normalsterbliche schier unendlich groß scheinender Pool an



Rechnern, aus dem man sich dynamisch bedienen kann. Das Mittel dazu ist die mittlerweile zur Commodity gewordene Virtualisierung, die Sie vom Einsatz im Clientumfeld, z. B. auf Basis von VMware oder Parallels kennen. Amazon bietet die Möglichkeit, über ein Web-Service-API oder darauf basierende Kommandozeilen- oder GUI-Anwendungen auf Basis einer Vielzahl vorgefertigter Images eine neue virtuelle Maschine (eine „Instanz“) anzulegen. Die dafür benötigte Zeit liegt in der Regel zwischen 60 und 120 Sekunden – egal, ob es sich bei dem verwendeten Image um einen rudimentären Debian-Linux-Server, eine Oracle-DB oder einen Windows-Server handelt. Virtualisierung ist natürlich im Serverumfeld nichts Neues: Schon lange wird diese z. B. auch von Internet Service Providern genutzt, um kostengünstig virtuelle Hosts anzubieten. Aber es gibt eine Reihe von Unterschieden zwischen einem solchen und dem EC2-Modell: Zum einen werden Sie eine Instanz noch schneller wieder los als Sie sie anlegen können, zum anderen zahlen Sie

nur für die Zeit, in der diese tatsächlich aktiv war. Darüber hinaus ist eine schier unüberschaubare Anzahl von Images verfügbar, die Sie um Ihre eigenen – öffentlichen oder privaten – erweitern können. Wir nutzen diesen Dienst in der Praxis für den kurzfristigen Einsatz von Testservern für unterschiedliche Kundenumgebungen oder für Demozwecke – von einem fertig konfigurierten und eingerichteten System wird quasi ein Schnappschuss erstellt und als Image für die spätere Verwendung in den ebenfalls von Amazon zur Verfügung

Rechnern, aus dem man sich dynamisch bedienen kann. Das Mittel dazu ist die mittlerweile zur Commodity gewordene Virtualisierung, die Sie vom Einsatz im Clientumfeld, z. B. auf Basis von VMware oder Parallels kennen. Amazon bietet die Möglichkeit, über ein Web-Service-API oder darauf basierende Kommandozeilen- oder GUI-Anwendungen auf Basis einer Vielzahl vorgefertigter Images eine neue virtuelle Maschine (eine „Instanz“) anzulegen. Die dafür benötigte Zeit liegt in der Regel zwischen 60 und 120 Sekunden – egal, ob es sich bei dem verwendeten Image um einen rudimentären Debian-Linux-Server, eine Oracle-DB oder einen Windows-Server handelt. Virtualisierung ist natürlich im Serverumfeld nichts Neues: Schon lange wird diese z. B. auch von Internet Service Providern genutzt, um kostengünstig virtuelle Hosts anzubieten. Aber es gibt eine Reihe von Unterschieden zwischen einem solchen und dem EC2-Modell: Zum einen werden Sie eine Instanz noch schneller wieder los als Sie sie anlegen können, zum anderen zahlen Sie

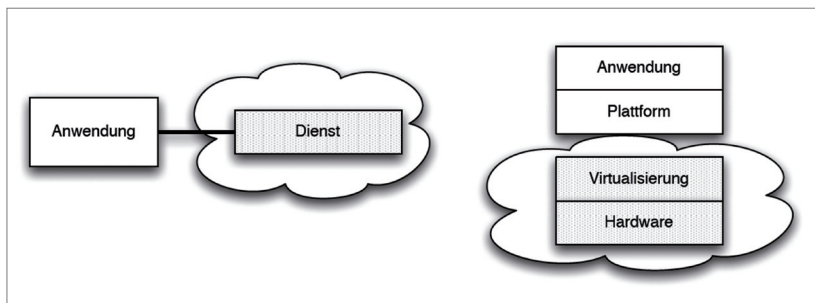


Abb. 1: Varianten des SaaS-Modells

gestellten, endlosen Speicherbereich S3 („Simple Storage Service“) gelegt.

Bei diesem Modell – das nur einen kleinen Ausschnitt der Amazon-Services darstellt (Kasten: „Amazon ANS“) – stellt ein Anbieter dem Anwender Infrastruktur zur Verfügung, die dieser dann beliebig nutzen kann. Das hat sowohl Vor- als auch Nachteile. Auf der einen Seite sind Sie völlig frei, was Sie mit Ihren Instanzen anstellen: Sie können ein mehr oder weniger beliebiges Betriebssystem installieren, Ihre eigene Datenbank- oder Applikationsserverversion aufspielen, Ihre möglicherweise unveränderte Anwendung dort in Betrieb nehmen usw. Der Nachteil ist, dass Sie genau das eben auch tun müs-

sen – Sie sind zwar von Themen wie Hardwarebeschaffung, Netzwerktopologie, SAN/NAS etc. befreit, aber es bleibt eine durchaus erhebliche Menge an Aufgaben, für die Sie selbst verantwortlich sind.

Das Modell, das Amazon mit EC2 anbietet, wird daher allgemein als „Infrastructure as a Service“ (IaaS) bezeichnet: Der Anbieter stellt Ihnen Infrastruktur zur Verfügung. Für die darauf basierende Plattform und die wiederum darauf laufenden Anwendungen sind Sie selbst verantwortlich. Neben EC2 hat Amazon noch eine ganze Reihe weiterer Dienste im Angebot, die über eine SOAP- oder HTTP-Schnittstelle (Letztere größtenteils nicht RESTful) genutzt

werden können, z. B. S3 für die Ablage von Daten oder CloudFront als Caching-Dienst. Nähere Informationen dazu finden Sie im Kasten. Diese Dienste lassen sich zum größten Teil ebenfalls auf der Infrastrukturebene ansiedeln.

Google App Engine: PaaS

Auch Google bietet eine Reihe von Infrastrukturdiensten an. Die meisten davon basieren auf dem Google Data API (GData), das seinerseits wiederum auf das REST-konforme Protokoll AtomPub aufsetzt – von SOAP und WSDL hat sich Google mittlerweile verabschiedet. Besonders interessant ist jedoch die Google App Engine (GAE), eine Ablaufumgebung für Anwendungen, die konform zu einem vordefinierten Programmiermodell implementiert werden müssen und dann in der Google-Infrastruktur ablaufen können. Für den Entwickler ist es dabei völlig unerheblich, ob die Anwendung auf einer oder einer Vielzahl von Maschinen, virtuell oder physisch abläuft – Google verspricht, eine Skalierung in nahezu beliebige Dimensionen zu gewährleisten.

Ursprünglich war eine GAE-Programmierungsumgebung ausschließlich für die Programmiersprache Python verfügbar. Mittlerweile gibt es auch eine Java-Version des GAE-Toolkits [1]. Als Java-Entwickler können Sie damit im Prinzip eine Webanwendung implementieren, die Ablaufumgebung entspricht Java SE mit dem Servlet-API. Für bestimmte Aspekte schreibt Google allerdings die Nutzung der eigenen Dienste vor: So ist zum Beispiel für die Persistenz BigTable verantwortlich, der hochskalierbare Datenspeicher, der von Google auch intern verwendet wird. Dieser ist entweder über ein Low-Level-API oder über eine nahezu vollständige JPA- oder JDO-Schnittstelle ansprechbar. Weitere Dienste, die Ihnen als GAE-Entwickler zur Verfügung stehen, sind „URL Fetch“ für den Aufruf externer HTTP-Services, XMPP für die Peer-to-Peer-Kommunikation, ein Dienst für die Manipulation von Bildern, ein verteilter Cache, die Möglichkeit zum Versenden von E-Mails und schließlich der TaskQueue-Mechanismus für die Unterstützung asynchroner Verarbeitung.

Die Google App Engine definiert damit ähnlich wie ein J2EE- oder Java-

Amazon AWS

Amazon kann man ohne Zögern als Pionier in Sachen Cloud Computing bezeichnen – kaum ein anderes Unternehmen stellt mit solcher Regelmäßigkeit neue Dienste zur Verfügung, die sich allesamt durch hohe Qualität und Verfügbarkeit auszeichnen. Ein Ausschnitt:

- Elastic Computing Cloud (EC2) ist eine Umgebung, in der auf Basis vorgefertigter oder eigener Images neue virtuelle Rechnerinstanzen in verschiedenen Ausbaustufen angelegt werden können.
- S3, der Simple Storage Service, erlaubt die Ablage von unstrukturierten Daten unter einem Key.
- Elastic Block Storage (EBS) bietet virtuelle Partitionen, die mit einer EC2-Instanz verknüpft werden können. Im Gegensatz zu den Inhalten der Festplatten, die direkt zur Instanz gehören, bleiben in einem EBS-Volume gespeicherte Daten erhalten.
- SimpleDB ist ein Key Value Store, in dem strukturierte Daten abgelegt werden. Er stellt damit eine hochskalierbare Alternative zu RDBMS-Systemen dar.
- SQS, der Simple Queueing Service, ist eine Art global erreichbares MOM- (Message-oriented-Middleware-)System.
- Unter RDS, Relational Datastore, versteht Amazon ein RDBMS (MySQL), das in der Cloud betrieben wird.
- DevPay ist eine Infrastruktur, über die Kosten für die Nutzung von Cloud-Ressourcen Dritten in Rechnung gestellt werden können.
- CloudFront ist ein Content Delivery Network, d. h. ein Netz, über das Inhalte wie Bilder, Videos usw. geografisch verteilt werden, um die Zugriffszeiten für Endanwender zu minimieren.
- Mit Elastic Map Reduce unterstützt Amazon die Kombination des Open-Source-Frameworks Hadoop mit EC2, um eine massiv parallele Verarbeitung großer Datenmengen zu unterstützen.

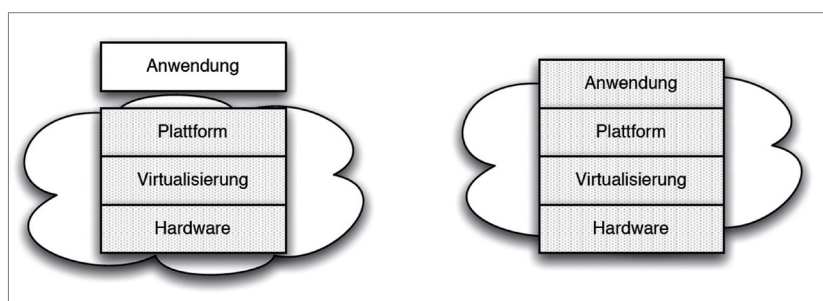


Abb. 2: PaaS- und SaaS-Modell

EE-Container einen Kontrakt zwischen der Anwendung bzw. deren Komponenten und der Ablaufumgebung, und ähnlich wie bei Java EE sind auch hier bestimmte Aktionen nicht erlaubt, z. B. der Zugriff auf das Dateisystem oder das Erzeugen von Threads. Im Gegenzug erhält man das Versprechen, dass sich die Ablaufumgebung um alles Wesentliche kümmert. Die lokale Entwicklung und den Test unterstützt Google mit einem SDK, das lokal auf dem Entwicklersystem installiert wird und von dem aus man die Anwendung (im Java-Fall per Ant-Script) in die Cloud deployen kann. GAE ist ein Angebot, das der Kategorie „Platform as a Service“ (PaaS) zuzuordnen ist – die Plattform läuft in der Cloud, als Endanwender kümmern Sie sich darum, eine plattformkonforme Anwendung zu entwickeln und zu deployen, für den Rest verlassen Sie sich auf die Kompetenz des Anbieters.

SaaS bei Salesforce.com und Google

Salesforce.com ist ein erfolgreicher Anbieter eines Customer-Relationship-Management-(CRM-)Systems, das vollständig webbasiert realisiert ist und dessen Installation für Unternehmen darin besteht, jeden zugriffsberechtigten Mitarbeiter online zu registrieren (Telefonnummer des Unternehmens in den USA: 1-800-NO-SOFTWARE). Dieses Modell wurde schon vor dem Aufkommen des Cloud-Computing-Begriffs als „Software as a Service“ (SaaS) bezeichnet – tatsächlich liegt darin die Historie der unsäglichen „As a Service“-Akronyme.

Auch Google bietet mit Google Mail, Docs, Calendar usw. eine ganze Reihe von Anwendungen an, die „in der Cloud“ ablaufen und damit dem SaaS-

Modell folgen. Auf die darin abgelegten Daten und teilweise auch deren Funktionen wiederum können Sie per API zugreifen – Sie werden Ihnen als Dienste in der Cloud zur Verfügung gestellt.

Abgrenzungen

Die Grenzen zwischen IaaS, PaaS und SaaS sind allerdings nicht so scharf, wie es die beschriebene Kategorisierung nahelegt. Amazon hat mit Elastic Map-Reduce einen Dienst im Programm, der verdächtig nach Plattform aussieht – hier können Sie einen parallelisierten Algorithmus, der eine große Datenmenge verarbeiten kann, über eine Konsole zur Ausführung registrieren. Zwar müssen Sie die Anzahl der zugrunde liegenden EC2-Instanzen noch selbst angeben, aber Sie übergeben Amazon eine .jar-Datei, die dort dann ausgeführt wird – die Amazon-Umgebung dient als Container, als Plattform für Ihre Anwendung.

Dass Google Service-APIs für die eigenen Anwendungen anbietet, haben wir schon erwähnt. Man kann darüber streiten, ob es sich auch beim Einsatz von webbasierten Anwendungen wie Google Mail oder Google Docs um Cloud Computing handelt – zumindest kann man in Frage stellen, was an diesem Modell neu ist.

Salesforce.com bietet mit Force.com auch eine PaaS-Lösung an, die noch eine Stufe oberhalb der Google App Engine anzusiedeln ist. Lösungen, die auf dieser Basis entwickelt werden, können nicht nur technische Dienste, sondern auch die fachlichen Funktionen der CRM-Lösung nutzen bzw. diese erweitern. Entwickelt wird dabei in einer eigenen, objektorientierten High-Level-Programmiersprache („APEX“), für die Salesforce.com eine Eclipse-basierte IDE ausliefert. Man könnte dies als eine

Analogie zu SAPs ABAP in der Cloud bezeichnen.

Cloud-Varianten

In den meisten Fällen bezieht sich der Begriff „Cloud Computing“ auf öffentliche Angebote, d. h. solche, die im Internet verfügbar und für jeden Inhaber einer Kreditkarte unmittelbar zugänglich sind. Zur Abgrenzung gegenüber alternativen Modellen wird diese Wolkenvariante häufig auch als *Public Cloud* bezeichnet.

Im Gegensatz dazu ist eine *Private Cloud* eine unternehmensinterne IT, die nach den gleichen Prinzipien organisiert ist. So nutzen zum Beispiel die Anbieter von Server-Virtualisierungslösungen den aktuellen Hype und etikettieren ihre Produkte entsprechend um. Die Herausforderung für eine Private Cloud besteht vor allem in der Umsetzung der entsprechenden Automatisierung – zumindest, wenn man die gleichen ehrgeizigen Ziele verfolgt wie die professionellen Anbieter: Eine Lösung aufzusetzen, in der das Anlegen einer neuen Serverinstanz nur wenige Minuten in Anspruch nimmt, ist nicht trivial. Auf der anderen Seite ist das möglicherweise in vielen Fällen gar nicht nötig – möglicherweise ist eine Wartezeit von einem Tag schon eine mehr als ausreichende Verbesserung.

Ähnlich wie der mittlerweile aus der Mode gekommene Begriff „Extranet“, der ein Netz bezeichnet, das weder öffentlich (wie das Internet) noch privat (wie ein Intranet) ist, ist eine *Community Cloud* eine Art Mittelweg: Kooperierende Organisationen, z. B. Unternehmen einer Branche oder Mitglieder eines Verbandes, schließen sich zusammen, um eine gemeinsame halböffentliche Cloud-Plattform zu betreiben.

Während eine Community Cloud eine Alternative zu Public und Private Clouds darstellt, ist eine *Hybrid Cloud* eine Mischung daraus – einige Teile einer Lösung werden z. B. in die Public Cloud ausgelagert, andere verbleiben im eigenen Rechenzentrum (oder zum Beispiel in einer klassischen, gehosteten Umgebung). Ein solches Modell wird sowohl von Amazon als auch von Google unterstützt. Beide Unternehmen bieten Lösungen an, bei denen ein VPN zwischen der Cloud-Plattform

und der unternehmensinternen IT die sichere Kommunikation unterstützt. So können z. B. in ein Google-Spreadsheet direkt die Daten aus einer unternehmensinternen Datenquelle importiert werden, und auch Anwendungen auf der Google App Engine können Legacy-Dienste im unternehmenseigenen Rechenzentrum aufrufen.

Chancen ...

Cloud Computing verspricht eine Reihe von Vorteilen. Auf die wichtigsten werden wir im Folgenden eingehen.

Kostenreduktion: Für die Verwalter der Finanzmittel sind die versprochene Kostenreduktion und die Umverteilung von Investitions- auf variable (operative) Kosten am wichtigsten. Die Begründung für Ersteres ist, dass ein Unternehmen mit einer eigenen IT-Infrastruktur diese für die Maximallast auslegen muss, auch wenn im Durchschnitt vielleicht nur ein Bruchteil davon benötigt wird. Da ein Cloud-Computing-Anbieter seine Ressourcen verschiedenen Mandanten zur Verfügung stellt, deren Auslastungsprofile sich stark unterscheiden können, steigt die Effizienz. Diesen Kostenvorteil kann der Anbieter an die Mandanten weitergeben (und tut dies hoffentlich auch). Auf der anderen Seite werden Investitionskosten reduziert: Stattdessen fallen laufende Kosten an, und diese idealerweise auch nur in einem der Nutzung angemessenen Maß.

Externes Know-how, Verfügbarkeit und Stabilität: Beim Einsatz einer Cloud-Lösung profitieren Sie vom Know-how des externen Anbieters. Bei Amazon oder Google steht deren Kompetenz im Betrieb hochskalierbarer Infrastrukturen wahrscheinlich außer Frage. Selbst bei den Angeboten, die zunächst nach „Low Level“ klingen, wie Amazons EC2, überlassen Sie es dem Anbieter, sich um Dinge wie Netzwerktopologie, Router und Switches, USVs, SAN und NAS usw. zu kümmern. Sie können davon ausgehen, dass Amazon und vergleichbare Anbieter diese Aspekte im eigenen Interesse hochoptimiert umgesetzt haben.

Automatisierung: Der gesamte Prozess für die Bereitstellung, aber auch für die Abschaltung eines Dienstes, eines

Infrastrukturelements oder einer Plattform ist hochgradig automatisiert. Für die Cloud-Anbieter ist dies eine Notwendigkeit – schließlich sprechen sie einen beliebig großen Kundenkreis an, der sich im Idealfall vollständig selbst versorgen soll. Dieses Modell steht im krassen Ge-

Cloud Computing bietet einige Vorteile, wie z.B. Kostenreduktion und Automatisierung.

gensatz zu den üblichen Vertriebsstrukturen, wie sie Infrastrukturlieferanten für ihre Großkunden aufbauen.

Dynamische Skalierbarkeit: Dieser Aspekt hängt mit den drei vorher genannten Aspekten eng zusammen und bezieht sich auf die Möglichkeit, Ressourcen je nach Bedarf dynamisch hinzuzufügen oder wieder zu entfernen. Ein viel zitiertes Beispiel für den Nutzen dieses Aspekts ist animoto.com. Dieser Online-dienst produziert aus Bildern, Musik und Text, die Sie hochladen, einen außerordentlich ansehnlichen Videoclip. Dieser Dienst wurde über eine Facebook-Applikation so populär, dass die Benutzerzahl von 25 000 auf 250 000 anstieg. Um diese Last zu bewältigen, konnte die Anzahl der Rechnerknoten (Amazon EC2, verwaltet über die Lösungen von Rightscale) kontinuierlich innerhalb dreier Tage von 50 auf über 3500 hoch- und nach einigen Wochen wieder auf mehrere Hundert herunterskaliert werden. Wo würden Sie 3450 Server innerhalb von drei Tagen beschaffen können?

... und Risiken

Aber natürlich ist auch in diesem Umfeld nicht alles perfekt. Sehen wir uns einige der am häufigsten genannten Bedenken gegen Cloud Computing – teilweise berechtigt, teilweise nur einfach immer wiederholt und dadurch nicht unbedingt überzeugender – näher an.

Zunächst einmal gibt es stellenweise *unklare und widersprüchliche Definitionen*. Dies ist nicht unbedingt ein Mangel am Ansatz, sondern an der Diskussion darüber – wie die Diskussion um das Thema SOA zeigt, spiegelt sich auch in

der Diskussion über Begrifflichkeiten die Gartner'sche Hypekurve mit den Phasen *übermäßige euphorische Erwartungen*, *extreme Desillusionierung* und *langfristige produktive Unaufgeregtheit* wider. In welcher Phase man sich befindet, ist sehr stark von den eigenen Erfahrungen abhängig.

Da das Thema noch relativ neu ist, gibt es einen *Mangel an Standards*, aus dem sich eine starke *Anbieterabhängigkeit* ergibt. So ist eine hochoptimierte Cloud-Anwendung nicht einfach vom einen zum anderen

Cloud-Anbieter portierbar. Allerdings gibt es dazu bereits einige interessante Ansätze, z. B. Eucalyptus, eine Open-Source-Implementierung des Amazon-EC2-APIs, oder die Bemühungen zur Standardisierung von Management-APIs von Sun. Aus dem geringen Alter des Ansatzes ergibt sich natürlich auch eine noch nicht unbedingt vertrauenerweckende *Produkt- und Dienstlebensdauer*. Hier sind Anbieter wie Amazon oder Google, die ihre eigene, auch selbst genutzte Infrastruktur als Basis für ihre Cloud-Angebote nutzen, klar im Vorteil.

Externes Know-how, Verfügbarkeit und Stabilität: Dieser Punkt lässt sich sowohl als Vorteil wie auch als Nachteil aufführen – für diese Aspekte ist man zwar nicht mehr selbst verantwortlich, hat sie aber auch nicht mehr selbst in der Hand. Auch hier unterscheiden sich die Anbieter deutlich und bieten zum Teil sehr ausgefeilte Garantien an, die einzuhalten für eine interne IT fast nicht möglich ist. Es prüfe, wer sich ewig bindet.

Sicherheitsaspekte: Dieses Themenfeld sorgt für die größten Zweifel am Ansatz. Vertraut man dem externen Anbieter nur Daten an, kann man diese vorher verschlüsseln – damit ist diese Nutzung eher unkritisch. Wenn aber die Verarbeitung der Daten auf den Systemen eines externen Partners erfolgen soll, gibt es keinen Mechanismus, mit dem man sie vor einem Zugriff durch Mitarbeiter des Anbieters mit Administrationsrechten schützen kann. Auf der anderen Seite kann man argumentieren, dass Unternehmen ohnehin schon heute ihre IT an Dritte outsourcen, häufig inklusive der Hardware und oft sogar mit

einer Übernahme interner Mitarbeiter durch den Outsourcing-Partner. Über Verträge wird geregelt, welche Garantien der Partner einhalten muss, um z. B. den Schutz von Daten vor dem Zugriff durch andere Mandanten – oft Mitbewerber – zu gewährleisten. Nichts anderes ist von einem Cloud-Anbieter zu erwarten.

Zum anderen ist die Frage berechtigt, wovon das größere Risiko ausgeht – von einer eigenen IT-Infrastruktur, die ein Unternehmen vor externen Angriffen schützen muss und auf die eigene Mitarbeiter intern administrativen Zugriff haben, oder von einer gut abgesicherten Google- oder Amazon-Infrastruktur.

Einsatzszenarien

Wenn Sie eine komplett neue Webanwendung erstellen, die im Internet verfügbar sein soll, sind PaaS-Angebote wie die Google App Engine eine hervorragende Alternative zum Aufbau einer eigenen Infrastruktur. GAE ist ebenfalls Mittel der Wahl für Widgets, Apps, Robots usw., die sich in Plattformen wie Facebook oder Google Wave integrieren sollen. Für Webanwendungen, die bereits implementiert sind, kann Amazon EC2 eine Alternative sein. In beiden Fällen profitieren Sie davon, dass Ihre Anwendung in einer Umgebung abläuft,

deren Skalierbarkeit wahrscheinlich über jeden Zweifel erhaben ist und für die Sie nur nach Bedarf zahlen müssen.

Wenn Sie eine einmalig oder selten zu erledigende Aufgabe haben, die sich parallelisieren lässt, können Sie die Cloud nutzen, um sich kurzfristig die benötigte Kapazität zu beschaffen. Hadoop bzw. Amazon Elastic Map/Reduce erlaubt Ihnen eine hocheffiziente Verarbeitung auf einer Menge von Systemen, die Sie wahrscheinlich nicht einfach in Ihrem Rechenzentrum für solche Fälle vorhalten können. Ein vielleicht nicht unmittelbar offensichtliches Beispiel für eine solche Aufgabe ist der Build-Prozess großer Enterprise-Systeme: Dieser lässt sich unter Umständen hervorragend parallelisieren, und Systeme wie Hudson unterstützen dies mittlerweile sogar ohne großen Aufwand.

Virtualisierungsmechanismen können für den Aufbau von Test- und Referenzsystemen eingesetzt werden, unabhängig davon, ob dies im Unternehmen oder in der Cloud geschieht. Im letzteren Fall profitieren Sie von einer Vielzahl bereits vorgefertigter, online verfügbarer Images und der direkten Anbindung ans Internet.

Schließlich lassen sich all diese Modelle auf unterschiedliche Art miteinander

der kombinieren, um spezifische Anforderungen zu unterstützen. So könnte eine Anwendung besonders sicherheitskritische Daten und Verarbeitungen intern durchführen, andere Verarbeitungsaufgaben über den Amazon-Queueing-Service SQS an einen auf EC2 laufenden, massiv parallelen Map/Reduce-Cluster weiterleiten, der seine Ergebnisse auf S3 ablegt. Von dort könnte eine Google-App-Engine-Anwendung die Daten abholen, um sie in einem skalierbaren Portal den Endanwendern darzustellen.

Fazit

Natürlich ist Cloud Computing ein Buzzword und daher mit Vorsicht zu genießen – nicht alles, auf dem das Etikett klebt, ist sinnvoll. Aber unabhängig von der Verwendung des Begriffs gibt es diverse Einsatzszenarien, in denen sich ein konkreter Vorteil ergibt, und eine ganze Reihe von Angeboten, die bereits heute dazu sinnvoll genutzt werden können. Als Architekt oder Entwickler stehen Sie vor der Herausforderung, sich mit den unterschiedlichen Architekturmodellen vertraut zu machen und zu beurteilen, welche davon für Ihr System sinnvoll sein können. Am einfachsten erschließt sich das Thema, wenn man die verschiedenen Angebote einfach ausprobiert. Bei Google und vielen anderen können Sie das kostenlos tun, bei Amazon kostet es Sie nur wenige Euro. Worauf warten Sie also? ■

Und was ist mit ...?

Utility Computing

Der Begriff „Utility Computing“ ist deutlich älter als Cloud Computing. Mit Utility ist in diesem Zusammenhang kein Werkzeug, sondern ein Versorgungsunternehmen wie z. B. ein Energie- oder Gasversorger gemeint, insbesondere dessen Abrechnungsmodell. Anstatt selbst in Generatoren, Turbinen oder ganze Kraftwerke zu investieren, stellen Versorgungsunternehmen Strom, Gas, Wasser usw. zur Verfügung und rechnen diese auf Verbrauchsbasis mit ihren Kunden ab. Ähnlich kann ein Unternehmen, anstatt selbst ein Rechenzentrum aufzubauen, einen Vertrag mit einem Rechenzentrumsanbieter abschließen und für die tages- oder monatsweise Nutzung von Systemen zahlen. Der wesentliche Unterschied zwischen Cloud Computing und Utility Computing liegt in der Virtualisierung – diese ist für Cloud Computing Voraussetzung, für Utility Computing nicht. Somit ist Cloud Computing eine Form von Utility Computing (aber nicht andersherum).

Grid Computing

Beim Grid Computing wird eine Vielzahl von unabhängigen, oft heterogenen Systemen zu einem Grid zusammengeschaltet, um eine massiv parallelisierte Berechnung durchzuführen. Beispiele dafür sind die Verarbeitung der Daten vom LHC des CERN oder das SETI-Projekt. Natürlich kann man Grid Computing in der Cloud betreiben, indem man z. B. eine Menge von Amazon-EC2-Instanzen mithilfe eines Frameworks wie GLOBUS zu einem Grid verbindet. Mehr verbindet Cloud und Grid Computing jedoch nicht.



Stefan Tilkov ist Geschäftsführer der innoQ Deutschland GmbH. Sein Fokus liegt auf Architekturen für stark verteilte Systeme und Systemlandschaften sowie effizienten Möglichkeiten zur Softwareentwicklung. Er ist Autor von „REST und HTTP“ (dpunkt 2009).

Links & Literatur

- [1] Google App Engine – Cloud Computing mit Java: www.jaxenter.de/artikel/2241