



Freund oder Feind?

Elastische Geschäfts- anwendungen mit dem Industriestandard

Torsten Fink, Florian Schneble

In der Cloud werden bei Bedarf Prozessoren und Speicher kostengünstig dazugeschaltet, ohne dass die Anwender Leistungseinbußen wahrnehmen. Ist der Ansturm vorbei, werden die zusätzlichen Ressourcen wieder abgeschaltet. Zusätzlich zu dieser Eigenschaft der Elastizität, ermöglicht die Cloud dem Kunden, maßgeschneiderte Anwendungsserver automatisch bereitzustellen, das sogenannte Software-as-a-Service. Für Architekten typischer Geschäftsanwendungen stellt sich nun die Frage, ob es sich für sie lohnt, Cloud-Technologien zu nutzen, und welche Risiken sie zu erwarten haben. Insbesondere stellt sich auch die Frage, ob der Industriestandard Java EE einsetzbar ist oder ob Cloud-spezifische Frameworks benutzt werden müssen.



Die Cloud – mehr als nur ein Betriebsthema

► Aus unserer Erfahrung wird das Thema Cloud immer wieder als ein reines Betriebsthema wahrgenommen. Schließlich ermöglicht es ein Infrastructure-as-a-Service (IaaS)-Anbieter, einfach neue Rechen- und Speicherressourcen ohne langfristige Vorausplanung zu erwerben und auch wieder abzustoßen. Diese Wahrnehmung unterschlägt die neuen Möglichkeiten, die für die Anwendungsentwicklung entstanden sind. Wir sehen hier zwei neue Anwendungsklassen:

- ▼ In einer Cloud können elastische Anwendungen sich entweder durch externe Konfiguration oder sogar automatisch an Hoch- und Niedriglastsituationen anpassen.
- ▼ Kunden können sich selbst in einer Webanwendung eine Anwendung inklusive Server konfigurieren und bezahlen. Der entsprechend zusammengestellte Applikationsserver wird dann automatisch in der Cloud bereitgestellt.

Diese Anwendungsklassen sind nur mit modernen Virtualisierungstechnologien und programmierbaren Verwaltungsschnittstellen zu realisieren. Beides gehört zu dem Angebot eines Cloud-Anbieters. Der direkte und automatisierbare Kontakt zum Kunden bietet neue technische Möglichkeiten und so neue Geschäftsmodelle.

Risiken und Nebenwirkungen

In Kongruenz zum Energieerhaltungssatz aus der Physik bringen neue Möglichkeiten in der Informationstechnik auch stets neue Herausforderungen mit sich. Um die Möglichkeiten eines Cloud-Anbieters ausnutzen zu können, stellt dieser meist proprietäre Schnittstellen zur Verfügung. Die strategische Freiheit, nutzungsabhängig IT-Ressourcen zu erwerben, wird so durch eine Bindung an den Cloud-Anbieter bezahlt, dem gefürchteten Vendor-Lock-in.

Für den Schutz von Daten und Systemen in einem internen Netzwerk existieren bekannte Standardmechanismen. Zu diesen gehören Firewalls, Verschlüsselung und Zugriffskontrollmechanismen sowie auch organisatorische Maßnahmen, wie Sicherheitsschleusen. Nutzt man allerdings die Infra-

struktur in einer Cloud, muss man den Sicherheitsmechanismen des Cloud-Anbieters vertrauen. Eventuell werden die Ressourcen sogar in einem fremden Land verwaltet und unterliegen einer anderen Gesetzgebung. Zusätzlich ist die Anbindung der Cloud-Ressourcen an interne Ressourcen über komplexe VPN-Mechanismen (Virtual Private Network) abzusichern.

Die Kopplung interner Ressourcen mit der Cloud bietet weitere Herausforderungen bzgl. Zuverlässigkeit und Leistungsfähigkeit. Der verbindende Netzwerkkanal ist sowohl Engpass als auch Single-Point-of-Failure. Ein Beispiel für eine äußerst ungünstige Konfiguration ist das Verlagern des zentralen Datenbanksystems in die Cloud. Alle internen Systeme sind dann von einer externen Ressource abhängig.

Wie nun vorgehen?

Der Einsatz von Cloud-Technologien birgt also Potenziale und Herausforderungen. In der initialen Analysephase eines Projekts ist daher abzuwägen, welche der Vorteile nutzbar sind und welche der potenziellen Probleme bzgl. Sicherheit, Zuverlässigkeit und Skalierbarkeit auftreten können. Nur wenn die Vorteile die Nachteile überwiegen, sollte der Einsatz der Cloud beschlossen werden. Um eine Herstellerabhängigkeit zu vermeiden, sind bei der Implementierung möglichst standardisierte Produkte einzusetzen. Wenn diese nicht ausreichen, sind dennoch Technologien zu vermeiden, die nur von einem Cloud-Anbieter unterstützt werden.

Der Industriestandard für Geschäftsanwendungen im Java-Bereich ist die Java Enterprise Edition (Java EE). Es stellt sich daher die Frage, inwieweit sich die Java EE für den Einsatz innerhalb einer Cloud einsetzen lässt. Um diese Frage zu untersuchen, haben wir drei Anwendungsklassen getrennt betrachtet:

- ▼ datenbankzentrierte,
- ▼ nachrichtenzentrierte und
- ▼ rechenintensive Anwendungen.

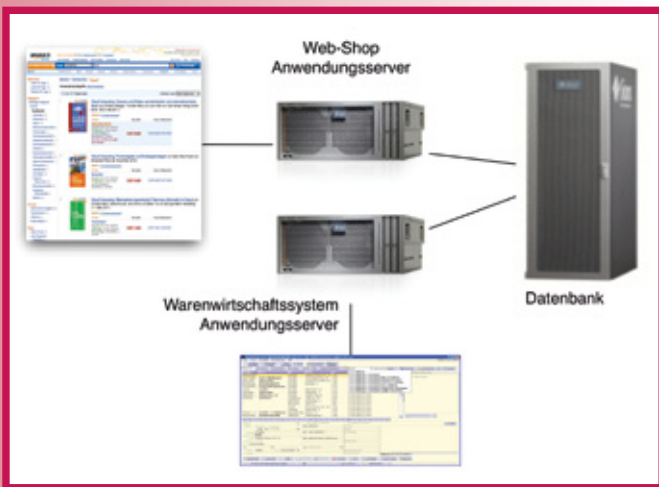


Abb. 1: Beispiel einer datenbankzentrierten Anwendung

Datenbankzentrierte Anwendungen

Typische Geschäftsanwendungen verwalten Stamm- und Bewegungsdaten, die in einem relationalen Datenbankverwaltungssystem (RDBMS) gehalten werden. Abbildung 1 zeigt eine beispielhafte Anwendung inspiriert von der Verkaufsplattform Amazon. Ein Webshop bietet Produkte im Internet an. Mit einem Warenwirtschaftssystem wird dann die nachfolgende Auftragsbearbeitung organisiert. Beide Anwendungen speichern ihre Daten in einem RDBMS.

Eine Systemoperation besteht meist aus der Übernahme von Eingabedaten aus der Benutzeroberfläche, dem Suchen und gegebenenfalls Verändern von Daten in der Datenbank und schließlich der Rückgabe und Aufbereitung der Ergebnisdaten. Die Hauptrechenlast entsteht im Allgemeinen auf dem RDBMS, welches die nebenläufig eintreffenden Such- und Änderungsanfragen koordinieren und abarbeiten muss.

Bei einer Umsetzung in einer Cloud ist primär darauf zu achten, dass der Kommunikationskanal zwischen dem Anwendungsserver und dem RDBMS zuverlässig und leistungsfähig ist. Dies bedingt im Allgemeinen, dass auch der Datenbankserver in die Cloud migriert werden muss. Je mehr Anwendungen auf diese Datenbanken zugreifen, desto mehr Anwendungen müssen entsprechend ebenfalls umgezogen werden.

Webanwendungen können sehr leicht in einem Cluster innerhalb einer Cloud betrieben werden. Dies ermöglicht eine gute Skalierbarkeit. Ist zusätzlich auch eine gesteigerte Zuverlässigkeit gewünscht, müssen die Daten zwischen den Clusterknoten repliziert werden. Hierfür bieten die einzelnen Java EE-Applikationsserver vielfältige Möglichkeiten an. Diese müssen aber aktuell noch in einer Cloud manuell konfiguriert werden, wofür eine hohe technische Kompetenz notwendig ist.

Um die Gesamtleistung des Systems zu erhöhen, ist die Leistung der Webanwendung häufig nicht so relevant. Wichtiger ist es, die Datenbankoperationen zu beschleunigen. Die meisten Cloud-Anbieter ermöglichen es, automatisch einem Server bei Hochlast mehr Prozessorkerne zuzuordnen. Da allerdings selten mehr als acht Kerne möglich sind, ist die Elastizität eingeschränkt. Das Aufsetzen von Datenbankclustern ist eine Kunst für sich, die noch nicht komfortabel von Cloud-Anbietern unterstützt wird.

Alternativ kann durch den Einsatz eines Zwischenpuffers der Zugriff auf die Datenbank reduziert werden. In Java EE-Anwendungen wird meistens über die Java Persistence API (JPA) auf ein RDBMS zugegriffen. JPA-Implementierungen,

wie Hibernate und EclipseLink, bieten hierfür sogenannte Second-Level-Caches an, die Lesezugriffe auf persistente Entitäten lokal bedienen können. Damit diese in einem Cluster zuverlässig und effizient funktionieren, ist wiederum eine technisch komplexe manuelle Konfiguration notwendig.

Eine weitere Alternative besteht in dem Einsatz eines NoSQL-Datenbanksystems. Es gibt verschiedene Klassen von NoSQL-Systemen. Geeignet für den Einsatz in Clustern sind auf jeden Fall alle Schlüssel-Wert-basierten Systeme, wie Apache Cassandra und Infinispan. Auch dokumentbasierte Systeme, wie CouchDB und MongoDB, sind geeignet. Da NoSQL-Datenbanken deutlich eingeschränkte Möglichkeiten für die Datenmodellierung und für Suchanfragen bieten, ist eine Umstrukturierung der Anwendungsarchitektur notwendig. Insbesondere für langlebige Stammdaten sollte der Einsatz einer NoSQL-Datenbank genau überlegt werden. Die Erfahrung zeigt, dass Daten häufig zu späteren Zeitpunkten auch von anderen Anwendungen für neue fachliche Operationen benötigt werden. Der große Vorteil von RDBMS im Vergleich zu NoSQL-Systemen besteht darin, dass sie meistens auch solche Anfragen effizient bearbeiten können, die bei der Datenmodellierung nicht berücksichtigt wurden.

Zusammenfassend kann man sagen, dass kleine und mittlere datenbankzentrierte Anwendungen mit entsprechendem Mengengerüst, die insbesondere nur aus einer Datenbank und einer Anwendung bestehen, sich gut in einer Cloud betreiben lassen. Bei diesem Szenario empfiehlt es sich, sowohl das RDBMS als auch den Applikationsserver auf einem elastischen Cloud-Server zu betreiben. Wenn höhere Anforderungen an Anzahl und Antwortzeitverhalten gleichzeitiger Zugriffe bestehen, dann findet sich aktuell in einer Cloud-Umgebung nur wenig Unterstützung.

Nachrichtenzentrierte Anwendungen

In dieser Anwendungsklasse fließen Daten per Nachrichten asynchron zwischen verschiedenen Anwendungsknoten. Abbildung 2 zeigt ein Beispiel aus der Telematik, eine Verwaltung von Fahrzeugen. Hierbei fließen Daten in zwei Richtungen. Die Fahrzeuge senden ihre Betriebs- und Nutzungsinformationen zu einer zentralen Flottenverwaltung. Diese speichert sie in einer Datenbank und bietet dem Anwender eine Auswertung in Form grafischer Berichte an. Umgekehrt kann ein Fuhrparkleiter die Zugänge zu den Fahrzeugen konfigurieren. Die entsprechenden Konfigurationsnachrichten werden an die Fahrzeuge gesendet und dort eingespielt.

In diesem Beispiel gibt es zwei alternative Kommunikationswege zu den Fahrzeugen. Zum einen können Fahrzeuge direkt per GPRS mit einem Kommunikationsserver kommunizieren, der ein Routing und eine Normalisierung von Nachrichten vornimmt. Zum anderen können Nachrichten über einen Datensammler offline per Bluetooth zu den Fahrzeugen übertragen werden. Auch hier dient der Kommunikationsserver als Nachrichtenweiche.

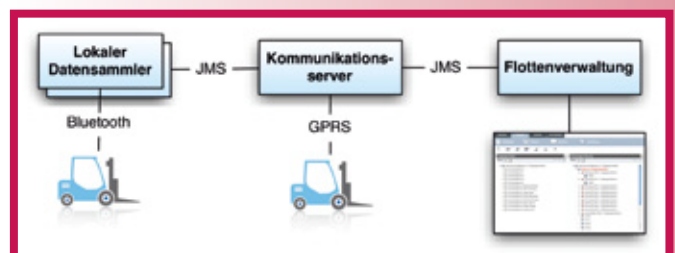


Abb. 2: Beispiel einer nachrichtenzentrierten Anwendung



Neben telematischen Anwendungen finden sich Beispiele dieser Anwendungsklassen auch bei Finanzdienstleistern, bei denen im Prinzip Geldbeträge per Nachrichten übertragen werden. Die wichtige Metrik zur Bestimmung der Leistungsfähigkeit ist hier der durchschnittliche und maximale Durchsatz des Systems. Ist dieser zu klein, führt dies im schlechtesten Fall dazu, dass die Nachrichtenpuffer beliebig groß werden. Der Durchsatz ergibt sich als Kombination aus der Bandbreite der Kommunikationskanäle und der Verarbeitungsgeschwindigkeiten einzelner Nachrichten.

Wenn der Cloud-Anbieter die eingesetzte Middleware unterstützt, ist eine Migration einfach. Für Java EE-Anwendungen bietet sich der Einsatz eines Nachrichtendienstes mit JMS-Schnittstelle an, wie z. B. ActiveMQ und HornetQ. Im Allgemeinen stützen sich Middlewareprodukte auf TCP/IP ab, das natürlich auch innerhalb von Clouds unterstützt wird. Dynamische IP-Adressen und VPN-Konfigurationen können allerdings Probleme bereiten. Multicast wird unseres Wissens noch gar nicht unterstützt.

Nachrichtenzentrierte Anwendungen bieten häufig eine gute Skalierbarkeit, da sich asynchrone, durch Nachrichten angestoßene Operationen in einem Cluster gut verteilen lassen. Java EE-Nachrichtendienste sind im Allgemeinen auch direkt Cluster-fähig. Die Konfiguration des Clusters ist allerdings wiederum technisch anspruchsvoll. Eine komfortable automatische Unterstützung eines Cloud-Anbieters ist uns nicht bekannt.

Rechenintensive Anwendungen

Klassischerweise versteht man unter rechenintensiven Anwendungen aufwendige numerische Berechnungen, wie z. B. Strömungssimulationen oder Auswertungen umfangreicher Sensordaten. Diese Applikationen zählen nicht zu den Geschäftsanwendungen und setzen spezielle Frameworks und Programmiersprachen ein, wie z. B. Message Passing Interface (MPI), High Performance Fortran und OpenMP.

Allerdings werden heute auch an klassische Anwendungen mit einer Onlinetransaktionsverarbeitung hohe funktionale Anforderungen gestellt. Eingehende Daten sind in Echtzeit aufzubereiten und bereitzustellen. Moderne Verkehrsleitsysteme empfangen z. B. pro Minute Betriebsdaten von über 1000 Fahrzeugen. Diese müssen ausgewertet, abgelegt und in dem Verkehrsleitstand angezeigt werden. Um derartige Anforderungen an die Antwortzeiten gewährleisten zu können, ist auch eine Parallelverarbeitung notwendig.

Java EE bietet mit JMS die Möglichkeit, Berechnungen innerhalb eines Clusters zu verteilen. Enterprise Java Beans (EJB) bietet zusätzlich ab der Version 3.1 die Möglichkeit an, lokale klassische parallele Berechnungen durchzuführen. Damit enthält Java EE das grundsätzliche Handwerkzeug, um auch rechenintensivere Aufgaben umzusetzen.

Wie schon erwähnt, unterstützen Cloud-Anbieter eine automatische Skalierung von Servern auf bis zu acht Kernen. Dies ermöglicht eine Skalierung der lokalen Berechnungen. Das Aufsetzen eines JMS-Clusters in einer Cloud ist möglich, benötigt aber eine manuelle Konfiguration. Damit können rechenintensiverer Java EE-Anwendungen mittlerer Größe auch in der Cloud laufen. Für anspruchsvolle Aufgaben sollten aber die nativen Möglichkeiten der Cloud-Anbieter genutzt werden. Falls möglich ist auf solche Programmierschnittstellen zu verzichten, die unter der direkten Kontrolle des Cloud-Anbieters liegen. Amazon unterstützt z. B. mit Apache Hadoop eine Open-Source-Schnittstelle für rechenintensive Anwendungen.

IaaS, PaaS oder SaaS

Zusätzlich zu der technischen Architektur stellt sich die Frage, welcher Cloud-Typ für die Entwicklung interessant ist. *Software-as-a-Service (SaaS)* ist das Ergebnisprodukt einer Cloud-Entwicklung. Im Rahmen der Entwicklung selber sind daher vor allem die SaaS-Angebote für eine Entwicklungsunterstützung interessant. So bietet die Firma JFrog z. B. ein Cloud-basiertes Maven-Repository an.

Infrastructure-as-a-Service (IaaS) ist aus Entwicklerperspektive wie ein ausgelagertes Rechenzentrum zu betrachten. Grundlegende Administratorkenntnisse sind zwar notwendig, allerdings müssen keine komplexeren Virtualisierungskompetenzen aufgebaut werden. Insbesondere lassen sich mit einer IaaS kostengünstig und schnell Test- und Abnahmesysteme bereitstellen.

Ein *Platform-as-a-Service (PaaS)*-Anbieter, der eine Java EE-Plattform zur Verfügung stellt, wäre natürlich ideal. In diesem Fall sind nur minimale Administratorkenntnisse notwendig. Aktuell bietet unseres Wissens nur die Firma Red Hat mit dem Angebot OpenShift eine solche Plattform. Die Basis ist hier der JBoss-Applikationsserver. OpenShift ist ein Vermittler zu weiteren Cloud-Anbietern, die dann die eigentlichen Rechen- und Speicherressourcen bereitstellen. Aktuell wird allerdings nur Amazon EC2 als Backend angeboten. OpenShift befindet sich noch in einer experimentellen Beta-Phase.

Des Weiteren hat Oracle eine Cloud-Plattform angekündigt. Diese enthält neben SaaS-Angeboten, z. B. für die Kunden- und Kontaktverwaltung (CRM), auch PaaS-Angebote für die Java EE-Plattform. Grundlage hierfür ist der WebLogic-Server. Für Entwicklungsabteilungen, die sich strategisch für die Oracle-Suite entschieden haben, zeichnet sich hier eine interessante Möglichkeit zur Entwicklung in der Cloud ab.

Resümee

Wie mit allen neuen Dingen bietet die Cloud Risiken und Potenziale. Insbesondere aber die Möglichkeit, neue Anwendungsklassen umsetzen zu können, macht ihren Einsatz attraktiv. Die Java Enterprise Edition kann im Cloud-Umfeld eingesetzt werden. Je nach Anwendungsklasse kann dies aber beliebig komplex werden. Für typische isolierte datenbankzentrierte Anwendungen mit mittlerem Lastgerüst ist die Cloud aber schon jetzt eine gute Wahl.



Dr. Torsten Fink ist Geschäftsführer der akquinet tech@spree und Leiter des JBoss Competence Centers. Neben der Leitung von Java EE-Projekten führt er Architektur- und Technologieberatungen durch.
E-Mail: torsten.fink@akquinet.de



Florian Schneble ist Mitarbeiter im JBoss Competence Center der akquinet. Er hat sein Informatikstudium an der HTW Dresden mit einer Diplomarbeit über das Thema „Einflüsse von Cloud-Konzepten auf unterschiedliche Architekturklassen von Geschäftsanwendungen unter Berücksichtigung der Java EE“ abgeschlossen.
E-Mail: florian.schneble@akquinet.de