Lothar Wieske

(E-Mail: lothar.wieske@deutschebahn.com)
ist Unternehmensarchitekt für Innovationsmanagement bei der DB Systel GmbH (ICT Tochter im Deutsche Bahn Konzern). Vorher hat er u. a. im Consulting für IBM, Microsoft und Sun Microsystems gearbeitet und schöpft aus Erfahrungen in Finance, Health und Transport & Logistics. Er beschäftigt sich schwerpunktmäßig mit Cloud Computing und ist Autor eines Fachbuchs.

Über den Wolken

Dateisysteme und Datenbanken im Cloud Computing

Cloud Computing transformiert die Informationstechnik. Die Konzepte zum Storage fallen bei den Anbietern von Infrastructure as a Service und Platform as a Service einstweilen noch recht unterschiedlich aus. Ein Überblick über Blob, Block, Queue und Table Storage stellt die Möglichkeiten vor. In Minutenschnelle zur relationalen Datenbank mit automatischem Backup- und Patch-Management? Auch dazu gibt es einen Einstieg. Die Erläuterungen des theoretischen Hintergrunds des CAP-Theorems und der praktischen Nutzung von Eventual Consistency machen Sie fit, sich mit Amazon Dynamo, Google Bigtable und Apache Cassandra zu beschäftigen.

Digitales Universum

Seit einigen Jahren beauftragen die Speicherspezialisten von EMC jährlich die Analysten von IDC mit der Vermessung des expandierenden digitalen Universums. Die aktuelle Studie vom Mai 2010 prognostiziert für das Jahr 2020 ein Anwachsen der digitalen Informationen auf ein unvorstellbares Volumen von 35 Zettabytes (ZB) – das ist eine Eins mit 21 Nullen [EMCIDC]. Einen Trend haben die Sternengucker ebenfalls ausgemacht: Während der Content – also die Masse im digitalen Universum – um den Faktor 44 zunimmt, steigt die Zahl der Container – also die stellaren Objekte wie Dateien, Nachrichten, Signale – sogar um den Faktor 69.

Für das Jahr 2010 mit einem Gesamtzuwachs von 1,2 Zettabytes steht

- einem Anteil von ca. 900 Exabytes an *User Generated Content*
- ein Anteil von ca. 260 Exabytes an *Enterprise Generated Content*

gegenüber. Übrigens, wer der stetigen Ausdehnung des digitalen Universums zuschauen möchte, kann das mit dem „Worldwide Information Growth Ticker“ unter [EMCDU].

Ein größerer Teil dieses User Generated Content läuft früher oder später über die technische Infrastruktur von Unternehmen und damit müssen sich die Unternehmen darum kümmern. Dieser sogenannte *Enterprise Touch Content* übersteigt mit ca. 960 Exabytes deutlich den Umfang des Enterprise Generated Content und mit ca. 600 Exabytes entfällt ein Löwenanteil des User Generated Content auf diesen Enterprise Touch Content. Man kann also sagen: In Unternehmen entstehen ca. 20 % der Inhalte des digitalen Universums. Unternehmen managen aber ca. 80 % der Inhalte des digitalen Universums.

Im Jahr 2010 sollen von den 35 Zettabytes etwa 5 als Informationen in Public und Private Clouds liegen (ca. 15 %). Nimmt man temporäre oder transiente Informationen in webbasierten E-Mail-Systemen und Online-Festplatten hinzu, summiert sich der Umfang immerhin auf etwa 12 Zettabyte an *Cloud Touch Content* (ca. 33 %).

Speichertopologien

Direct Attached Storage (DAS, s. Abb. 1) erlaubt mit Punkt-zu-Punkt-Verbindungen zwischen einem Server und einem Speichergerät (Band, Platte) das Speichern großer Datenmengen. Konzepte wie *Network Attached Storage* (NAS, s. Abb. 2) und *Storage Area Network* (SAN, s. Abb. 3) binden mehrere Server an mehrere Speichergeräte an – über große Distanzen und bei hohen Übertragungsgeschwindigkeiten.

Abb. 1: Direct Attached Storage (DAS)

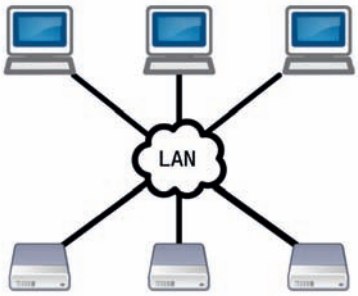

Abb. 2: Network Attached Storage (NAS)

Während der Server bei SAN auf Speicherblöcke zugreift (blockbasierend), ruft der Server bei NAS direkt Dateien oder deren Ausschnitte ab (dateibasierend). Ein NAS-Speichergerät lässt sich einfach in das vorhandene TCP/IP-Netzwerk einhängen und arbeitet mit den Protokollen *Common Internet File System* (CIFS), *Server Message*

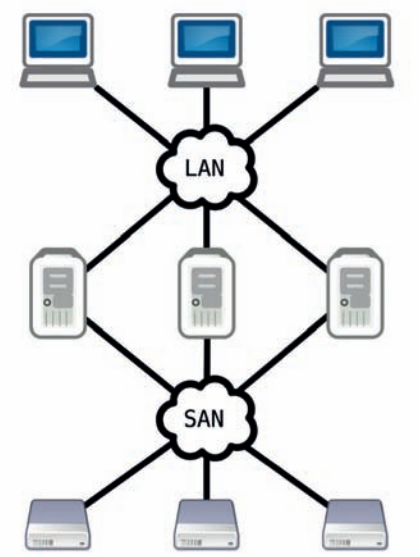

Abb. 3: Storage Area Network (SAN)

Block (SMB) sowie *Network File System* (NFS); die Dateizugriffe des Clients verwenden TCP/IP-Protokolle. Im SAN werden die Geräte in eigenständige Speichernetze mit *Fiber Channel* (FC), *Host Bus Adapter* (HBA) und *Switched Fabric*-Topologien eingebunden; die Blockzugriffe der Clients verwenden dann auch FC-Protokolle. Neuere SAN-Protokolle wie FCoE oder iSCSI erlauben auch blockbasierende Zugriffe über TCP/IP-Netzwerke. Und InfiniBand als serielle Übertragungstechnologie für hohe Geschwindigkeiten ist ebenfalls ein Kandidat im SAN und minimiert Latenzzeiten durch Auslagern des Protokoll-Stacks in die Netzwerkhardware. Nach diesem Schnelldurchlauf durch Technologie und Topologien zur Verbindung von Rechnern und Speichern stellt sich natürlich die Frage: Wie funktioniert das denn beim Cloud Computing und bei Infrastructure as a Service? An den Beispielen von Amazon, GoGrid und Rackspace wird klar, welche Konzepte umgesetzt werden und wie vielfältig die Landschaft sich derzeit darstellt.

Amazon – Elastic Compute Cloud

Die *Elastic Compute Cloud* (EC2) von Amazon bietet eine virtuelle Rechnerumgebung, in der eigene Instanzen über Webservices gestartet und verwaltet werden können. Das Starten einer Instanz erfordert wenige Schritte und/oder Operationen. Der Nutzer wählt aus einer Sammlung von Vorlagen ein vorkonfiguriertes Image mit vorinstalliertem Betriebssystem (Linux, Solaris, Windows) aus, entscheidet sich für die Ausstattung seines Rechners (32bit/64bit, Anzahl der Prozessorkerne, Hauptspeicher, Festplatte), nimmt Sicherheits- und Netzwerkeinstellungen vor und bestimmt ein Rechenzentrum für seine Instanz. Die Rechenzentren von Amazon sind in Regions und Availability Zones gegliedert. Derzeit gibt es zwei Regions in Amerika und jeweils eine Region in Asien und Europa; jede Region hat zwei bis vier Availability Zones.

Es gibt zwei Ablageorte für die Images, aus denen eine Instanz entsteht: *Simple Storage Service* (S3) und *Elastic Block Store* (EBS). Eine virtuelle Instanz aus S3 (Instance Store Image) erhält in EC2 einen flüchtigen virtuellen Storage. Er wird mit dem Starten der Instanz vergeben und belegt und mit dem Stoppen der Instanz auch wieder freigegeben und entfernt. Eine virtuelle Instanz aus EBS (*Block Store Image*) erhält in EBS einen beständigen virtuellen Storage, der auch nach dem Stoppen der Instanz erhalten bleibt. EBS Volumes werden gesondert in Rechnung gestellt.

Mit EBS kann ein Boot Device in eine Instanz eingehängt und/oder ein (weiteres) Block Device an eine Instanz angehängt werden. EBS Volumes werden in einer und für eine bestimmte Availability Zone erstellt und können eine Größe zwischen 1 GB und 1 TB haben. Nach der Erstellung kann ein Volume jeder Instanz in derselben Availability Zone zugewiesen werden. Anschließend wird das Volume wie eine Festplatte bzw. eine Plattenpartition angezeigt und kann formatiert und montiert werden.

Ein Volume kann jeweils nur einer einzigen Instanz zugewiesen werden. Es ist jedoch möglich, einer einzigen Instanz mehrere Volumes zuzuweisen.

Jedes Volume wird automatisch innerhalb seiner Availability Zone repliziert. Dadurch werden Datenverluste verhindert, die sich durch den Ausfall einzelner Hardwarekomponente ergeben könnten. Außerdem kann der Nutzer für ein Volume ein Snapshot erstellen, das in S3 gespeichert wird. Diese Snapshots werden automatisch in mehreren Availability Zones repliziert.

Bei Verwendung von EBS als Boot Device kann eine Instanz zeitweise angehalten und erneut gestartet werden. So zahlt der Nutzer während des Schönheitsschlafs nur für den Speicher und nicht für den Rechner. Außerdem kann der Nutzer beim Neustart eine veränderte Ausstattung beispielweise mit weniger Kernen und größerem Hauptspeicher wählen und damit in gewissen Grenzen skalieren. Für erhöhten Durchsatz, verbesserte Absicherung oder zum Durchbrechen der Beschränkung auf 1 TB können mehrere Volumes mit RAID (Software) gekoppelt werden.

Im Ergebnis bildet EC2 mit seinen Instance Store Images einen DAS-Ansatz ab und in seinen Block Store Images kommt ein SAN-Ansatz zum Tragen, der durch zusätzliche Volumes seine eigentlich Stärke im praktischen Einsatz zeigt. Flexible Nutzung mit einfacher Bedienung über Webservices und automatischer Sicherung mit Replikation in und über Availability Zones runden das Konzept ab.

GoGrid – Cloud Server und Cloud Storage

Auch GoGrid bietet mit seinen Cloud-Servern eine virtualisierte Rechnerumgebung – für Windows und Linux (CentOS, RHEL und demnächst Ubuntu). Die beiden GoGrid-Cloud-Rechenzentren befinden sich an der Ost- und Westküste Amerikas. Cloud-Server gibt es in sechs Größen: 0,5 (virtuelle Kerne)/0,5 GB (RAM)/30 GB (Storage), 1/1/60, 2/2/120, 4/4/240, 8/8/480 sowie 16/16/960. Rechnerleistung und Speichervermögen wachsen also linear gekoppelt.

Der Cloud Storage von GoGrid bietet dem Nutzer einen Service für das Sichern von Dateien auf seinen Cloud-Servern. Jeder Nutzer bekommt eine eigene Subdomäne <<customernumber>>.cloud.storage.gogrid.com zugewiesen und spricht diese in einem abgesicherten Subnetz an. Als Übertragungsprotokolle für die Dateiübertragung stehen rsync, ftp, samba und scp zur Auswahl.

GoGrid setzt mit seinen Cloud-Servern also einen DAS-Ansatz mit RAID-Unterstützung um und der Cloud Storage baut auf einen NAS-Ansatz.

Rackspace – Cloud Servers und Cloud Files

Rackspace baut seine virtualisierte Rechnerumgebung für Cloud Servers auf eine Xen-Plattform und bedient Linux mit CentOS/Fedora/Oracle EL/RHEL; Cloud Servers für Windows befinden sich in der Erprobung. Rackspace bietet Cloud Servers in drei seiner amerikanischen Rechenzentren (zwei in Texas, eins in Illinois) an. Es gibt sie in sieben Größen: 256 MB (RAM)/10 GB (Storage), 512/20, 1024/40, 2048/80, 4096/160, 8192/320 und 15872/620. Auch Rackspace koppelt die Dimensionierung der Speicherhierarchie. Die Wirtsysteme bei Rackspace haben Dual-Quad-Core-Prozessoren. Jeder Cloud Server als virtualisiertes

Gastsystem bekommt vier virtuelle Kerne zugewiesen und die Zahl der CPU-Zyklen wird entsprechend der Größe des Cloud Server gewichtet. Ein 4GB-RAM-Server erhält also eine doppelte Gewichtung gegenüber einem 2GB-RAM-Server.

Die Cloud Files bei Rackspace ermöglichen die Ablage von Dateien über einen einfachen Webservice (REST, Representational State Transfer). Die Dateien liegen größenmäßig maximal bei 5 GB und werden in Containern organisiert, die aber nicht hierarchisch geschachtelt werden (kein hierarchisches Dateisystem). Bei privaten Containern wird der Datenverkehr über SSL verschlüsselt. Bei öffentlichen Containern erhalten die Dateien eine URL für den Zugriff im Limelight Content Delivery Network.

Rackspace gibt seinen Cloud Servers ebenso einen DAS-Ansatz mit RAID-Unterstützung mit. Cloud Files sind ein eigenständiger Webservice.

Storage-Abstraktionen im Cloud Computing

Bei einschlägigen IaaS-Anbietern kommen die Ansätze DAS, NAS und SAN in der Virtualisierung also in ganz unterschiedlicher Weise zum Tragen. Schaut man sich Storage im Cloud Computing jenseits des Vergleichs mit bisherigen Konzepten an, so haben die Clouds in der Kombination Infrastructure as a Service (IaaS) und Platform as a Service (PaaS) vier Nutzungsformen für Storage hervorgebracht.

Block Storage

Für Infrastructure as a Service ist Block Storage eine zentrale Abstraktion. Bei Amazon zieht sich der Nutzer eine Compute-Instanz über die Elastic Compute Cloud (EC2) und eine Storage-Instanz über Elastic Block Storage (EBS). Dann verknüpft er beide und vergibt für die Verknüpfung einen Gerätenamen („/dev/sdf"). Mit diesem Gerätenamen kann die Storage-Instanz als Block Storage Device auf der Compute-Instanz formatiert und eingebunden werden („mkfs ... /dev/sdf / mount ... /dev/sdf").

Amazons EBS ist innerhalb von EC2 das typische Beispiel für Block Storage.

Blob Storage

Blob Storage speichert Binärdaten (Audio, Video, Word, Excel, PowerPoint, ...). Aus Nutzersicht sind das eigentlich Binärdateien. Aber Blob Storage bietet in der Regel kein Dateisystem, sondern nur ein oder mehrere Namensräume (Buckets, Container) mit einer einfachen und flachen Zuordnung von Schlüsseln zu Inhalten. Dieser Zuordnung wird über Webservices verwaltet (REST) und die Daten bekommen je nach Zugriffsrechten und Einstellungen eine URL, über die auch öffentlich auf die Daten zugegriffen werden kann.

Amazon S3 und Rackspace Cloud Files gehören ebenso in diese Kategorie wie auch Microsoft Windows Azure mit seinem Blob Storage und Google mit seinem Blob Storage für die Google App Engine.

Table Storage

Table Storage legt Entities in einer Tabelle ab, die im Unterschied zu einer relationalen Datenbank nicht über ein Schema definiert wird. Unterschiedliche Entities in der gleichen Tabelle fügen unterschiedliche Attribute hinzu und lassen sie weg oder verwenden sie mit unterschiedlichen Typen.

Bei Amazon heißt der Table Storage einfach SimpleDB, bei Microsoft Windows Azure wieder Table Storage und bei der Google App Engine trägt er die Bezeichnung Datastore.

Queue Storage

Queue Storage bietet einen Nachrichtenspeicher zur Unterstützung asynchroner Kommunikation, über den sich Sender und Empfänger austauschen können.

Amazon nennt seinen Queue Storage SQS für Simple Queuing Service, Microsoft Windows Azure hat Queue Storage im Angebot und bei der Google App Engine bietet die Task Queue eine vergleichbare Konstruktion.

Nutzung von Cloud Storage

Die Cloud-Services für Storage (Blob, Table, Queue) heißen also bei unterschiedlichen Anbietern ganz unterschiedlich und sind manchmal auch beschränkt auf den Zugriff innerhalb eines PaaS-Angebots (Google App Engine). Letztlich bieten sie eine Art Convenience Storage mit interessanten Leistungsmerkmalen, einfacher Verwendung und hoher Verfügbarkeit. In der Bereitstellung als eigenständiges IaaS-/PaaS-Angebot sind sie meistens mit REST-/SOAP-Schnittstellen ausgestattet, für die einfache Verwendung in Anwendungen wird gerne mit sprachspezifischen Bibliotheken oder Frameworks gekapselt.

Datenbanken im Cloud Computing

Nun erinnert Table Storage zwar dem Namen nach an die Tabellen relationaler Datenbanken – aber relationale Datenbanken bieten mit ihren Verknüpfungsmöglichkeiten und ihrer Transaktionsunterstützung mehr. Als PaaS-Angebote gibt es solche Datenbanken mit Amazons Relational Database Service und Microsoft SQL Azure und im IaaS-Bereich kommen vorbereitete Images mit installierter Datenbankmanagementsoftware zum Einsatz.

Amazon – Relational Database Service

Der Amazon Relational Database Service (RDS) gibt seinen Nutzern Zugriff auf MySQL-Datenbanken als Managed Service. Über einen Webservice werden diese Datenbanken erstellt, geändert, betrieben und gelöscht. RDS führt automatisch Patches der Software durch und erstellt Backups der Datenbank. Eine flexible Wahl und Änderung der Rechnerressourcen oder Speicherkapazitäten zwischen ein bis acht virtuellen Kernen, 1,7 bis 68 GB Hauptspeicher und 5 bis 1 TB Datenbankgröße ist ebenfalls möglich.

Microsoft - SQL Azure

Was Amazon mit RDS für MySQL anbietet, stellt Microsoft mit SQL Azure für den SQL-Server zur Verfügung. Es gibt zwei Ausgaben von SQL Azure. Die Web-Edition ermöglicht Datenbanken mit 1 oder 5 GB Speicherkapazität. Die Business-Edition erlaubt Datenbanken von 10 bis 50 GB in Zehnerschritten.

Amazon – AMIs für IBM, Oracle und Microsoft

Weiter oben war schon von Vorlagen für die Instanzen in der Elastic Compute Cloud von Amazon die Rede. Diese Vorlagen tragen die Bezeichnung *Amazon Machine Image* (AMI). Die großen Datenbankhersteller haben in Zusammenarbeit mit Amazon solche AMIs für ihre Datenbanken erstellt. In ihnen ist bereits die vollständige Datenbanksoftware installiert und der Nutzer kann nach dem Start der Instanz unmittelbar mit der Dimensionierung und Initialisierung der Datenbankinstanz loslegen. Neben der reinen Installation der Datenbanksoftware sind natürlich auch Umgebungen und Einstellungen bereits vorbereitet und werden herstellerseitig unterstützt (Kernel, Bibliotheken).

Und Datenbanken im Cluster?

Bisher ging es um einzelne Datenbanken. Datenbanken im Cluster sind eine eigene Betrachtung wert. Hier hat das Cloud Computing neuartige Entwürfe hervorge-

bracht und leistungsfähige Lösungen gebaut. Interessanterweise haben Unternehmen wie Amazon, Facebook, Google, Twitter und Co über ihre Ansätze publiziert oder die eigenen Entwicklungen kurzerhand als Open Source freigegeben. Deswegen wird sich der Folgeartikel mit Entwurfsmustern für hochverfügbare Datenbanken im Allgemeinen und Apache Cassandra als produktionsfähiger Implementierung im Besonderen beschäftigen.

Vorher soll diese praktische Begegnung mit der hochverteilten und hochverfügbaren Datenbank von Facebook (Cassandra) auf eine theoretische Grundlage gestellt werden.

CAP-Theorem: Consistency, Availability und Partition Tolerance

Im Jahr 2000 sprach Eric Brewer die Vermutung aus, dass sich die systemischen Merkmale Availability und Consistency bei verteilten Systemen gegenseitig ins Gehege kommen [Brew00]. Insbesondere unterschied er die Anwendungsklassen

- ACID (Atomicity, Consistency, Isolation, Durability) und
- BASE (Basically Available, Soft State und Eventual Consistency)

als widerstreitende Extreme in einem Kontinuum von Möglichkeiten. Der Architekt muss den pH-Wert seines „Entwurfs" in der Integration von ACID-Anteilen und BASE-Beiträgen einpendeln.

Im Kern seines Vortrags stand die Brewersche Vermutung (Brewer Conjecture) bezüglich Consistency, Availability und Partition Tolerance: *"You can have at most two of these properties in any shared-data system."* Im Jahr 2002 haben Seth Gilbert und Nancy Lynch vom MIT die Vermutung formal bewiesen und sie damit zum Brewer-Theorem oder auch CAP-Theorem (Anfangsbuchstaben der verknüpften Merkmale) erhoben [GiLy02].

Weil Partition Tolerance im Internetumfeld immer gefordert ist, bleibt eigentlich nur die Abwägung zwischen Availability und Consistency.

Im Sinne von Brewer, Gilbert und Lynch ist die Consistencyvon BASE eine Atomic Consistency. Diese atomare Konsistenz fordert eine totale Ordnung für alle Lese-/Schreib-Operationen. Und bezogen auf diese totale Ordnung soll jede Lese-Operation, die auf eine Schreib-Operation folgt, den Wert entweder dieser Schreib-Operation oder einer noch späteren Schreib-Operationen liefern. Also ist Atomic Consistency die Eigenschaft einer einzelnen Operation. Im Gegensatz dazu beziehen sich Database Atomicity und Database Consistency auf Eigenschaften ganzer Folgen von Operationen. Obwohl gleiche Begriffe verwendet werden, sind die dahinterliegenden Ansprüche stark unterschiedlich.

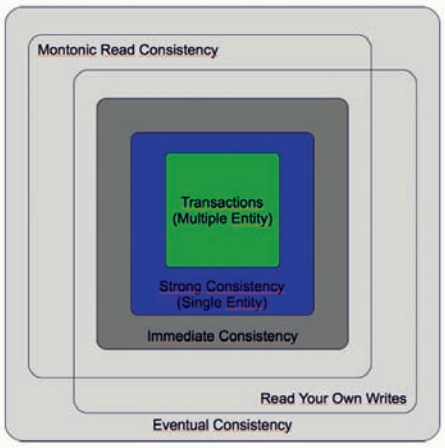

Abb. 4: Eventual Consistency

Eventual Consistency

Weiter oben war von einem Kontinuum die Rede und dieses Kontinuum möglicher Konsistenzgarantien ist stark verschachtelt. Vielleicht erhellt Abbildung 4 die Möglichkeiten etwas.

Es gibt also mit Eventual Consistency ein übergreifendes und fein abgestuftes Kontinuum mit Strong Consistency für einzelne Entities und Transactions für mehrere Entities als kleinsten Teilmengen.

Im Cloud Computing spielt Eventual Consistency eine wichtige Rolle [Vog08]. Mit Apache Dynamo legen die Kunden des Internethändlers jederzeit Waren im Einkaufskorb ab. Das verteile Google File System ist für die Internetsuche mit hohen Datendurchsätzen optimiert. Und bei Facebook unterstützt Cassandra in einem Cluster mit mehr als 600 Prozessorkernen und einem Plattenumfang von mehr als 120 TB die Inbox Search.

Amazon Dynamo, Google File System und Apache Cassandra sind Beispiele für verteilte Systeme mit Eventual Consistency.

Zusammenfassung

Ein Schnellüberblick über Topologien für Storage (DAS, NAS, SAN) stellte deren Umsetzungen bei den IaaS-Angeboten von Amazon, GoGrid und Rackspace gegenüber. Schlüsselfertige Services für Blob, Queue und Table Storage können über ihre REST-Schnittstellen flexibel in Anwendungen eingebunden werden und bieten verbrauchsbezogene Abrechnung bei hoher Verfügbarkeit.

Auch relationale Datenbanken mit automatisiertem Backup- und Patch-Management sind im Angebot und vereinfachen die Arbeit für Entwicklung und Betrieb. Das CAP-Theorem und Eventual Consistency bilden die Grundlage für innovative Konzepte für Dateisysteme und Datenbanken in den Clouds von Amazon, Google und Facebook. In einem Folgeartikel werden die Konzepte von Amazon Dynamo, Google File System, Google Bigtable am Beispiel von Apache Cassandra veranschaulicht und praktisch untersucht. ■

Literatur

[Brew00] E. A. Brewer, Towards Robust Distributed Systems, PODC Keynote, 19.7.2000, http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf

[EMCIDC] IDC Go-toMarket Services, http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm

[EMCDU] About EMC: Leadership and Innvoation: The Digital Universe, http://www.emc.com/leadership/digital-universe/expanding-digital-universe.htm

[GiLy02] S. Gilbert, N. Lynch, Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services, http://lpd.epfl.ch/sgilbert/pubs/BrewersConjecture-SigAct.pdf

[Vog08] W. Vogel, Eventually Consistent – Revisited, http://www.allthingsdistributed.com/2008/12/eventually_consistent.html