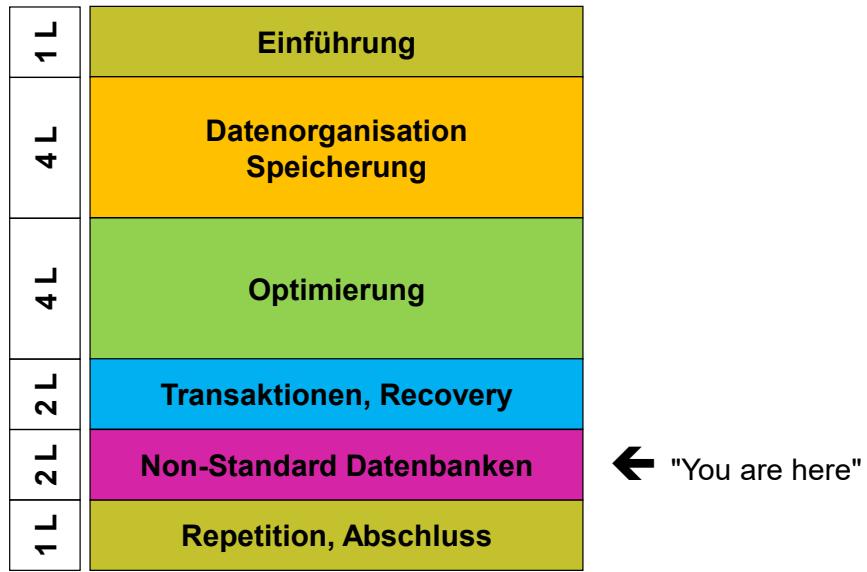


Lektion 13: Big data, NoSQL

Gliederung



2

- Operative versus dispositive Systeme
- Begriff und Architektur von data warehouses
- Mehrdimensionale Datenmodellierung:
 - Fakten
 - Dimensionen
 - Operationen
 - Drill-down, roll-up, slice & dice
 - Sternschema

Lernziele heute

Zürcher Hochschule
für Angewandte Wissenschaften



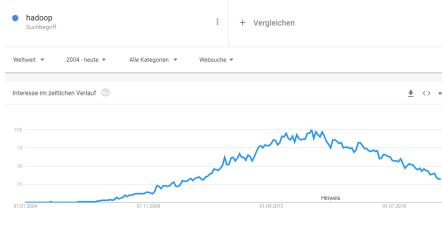
- Begriff big data kennen.
- Grenzen relationaler Technologie beurteilen können.
- Ansätze kennen, die unter dem Stichwort NoSQL beschrieben werden.

4

Zürcher Fachhochschule

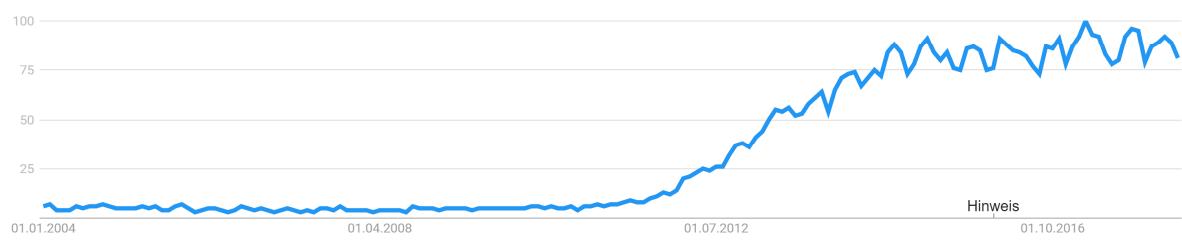
Big Data und verwandte Termini

Seit 2011 ist ein grosser Anstieg bei „Big Data“ zu erkennen:



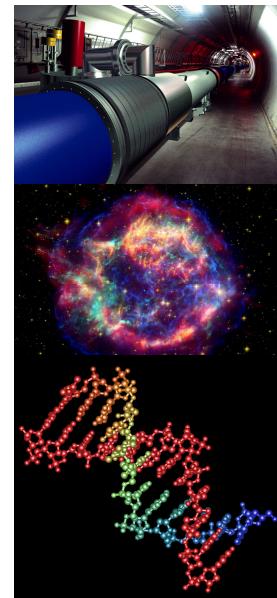
5

Zürcher Fachhochschule



Wichtigste «Wellen» des Datensunamis

- Welle 1 – Grossforschungsanlagen:
 - z.B. CERN, ...
- Welle 2 – Web-Firmen:
 - z.B. Amazon, eBay, Google, Yahoo, ...
- Welle 3 – Soziale Netzwerke:
 - z.B. Facebook, Twitter, LinkedIn, ...
- Welle 4 – Maschinen-generierte Daten:
 - z.B. Mobiltelefon, Industrie 4.0, IoT, ...
- Welle 5 ???



Was ist «big»?

Zürcher Hochschule
für Angewandte Wissenschaften



- 1 Terabyte ($1'000^4$)?
 - 300 Stunden hochauflösendes Video
- 10 Terabytes?
 - Alle Bücher der Library of Congress
- 1 Petabyte ($1'000^5$)?
 - Daten einer Grossbank
- 100 Petabyte?
 - Daten des CERN (15 Petabytes pro Jahr)

7

Zürcher Fachhochschule

90% aller Daten sind in den letzten beiden Jahren generiert worden

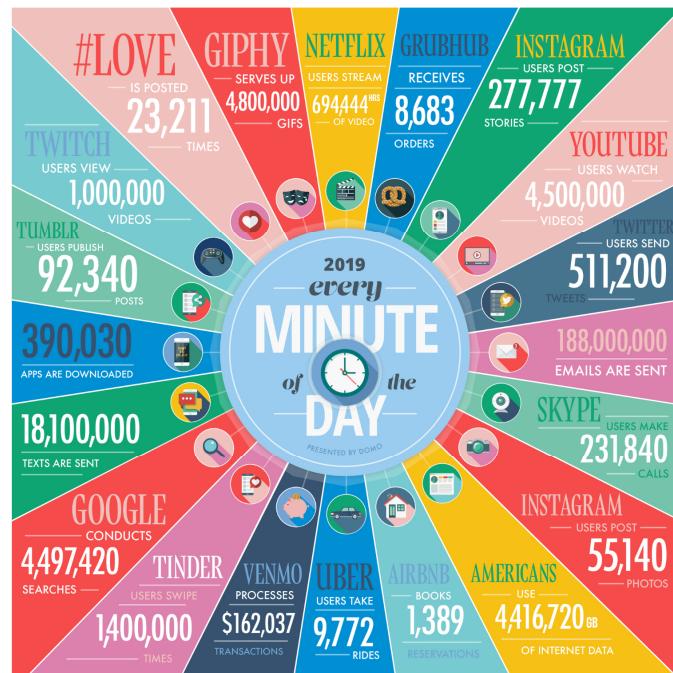
- Kilo: $1'000$
- Mega: $1'000^2$
- Giga: $1'000^3$
- Tera: $1'000^4$
- Peta: $1'000^5$
- Exa: $1'000^6$
- Zetta: $1'000^7$
- ...

Erzeuge Datenmenge CERN pro Sekunde (je Experiment):

- ATLAS: 320 Megabyte / Sekunde
- CMS: 220 Megabyte / Sekunde
- ALICE: 100 Megabyte / Sekunde
- LHCb: 50 Megabyte / Sekunde

Was ist «big»?

How much data is
generated every
minute? (2019)



<https://www.digitalinformationworld.com/2019/07/data-never-sleeps-7-infographic.html>

Zürcher Fachhochschule

- Die 5V's:
 - **Volume:** Grosses Datenvolumen
 - Terabyte bis Petabyte
 - **Velocity:** Hohe Stream-/Verarbeitungsgeschwindigkeit
 - Viele Datensätze pro Tag/Stunde/Minute
 - **Variety:** Unterschiedliche Datenquellen
 - Texte, Bilder, Videos, Datenbanken, Data Warehouse, Blogs, Soziale Netzwerke etc.
 - **Veracity (Echtheit):**
 - Daten mit unterschiedlicher Qualität
 - **Value:**
 - Wert der Daten (monetär, nicht Inhalt)
- Herausforderungen:
 - Datenerfassung, Speicherung, Auswertung und Visualisierung.

9

Zürcher Fachhochschule

Diskussion – ist das eine gute Definition?

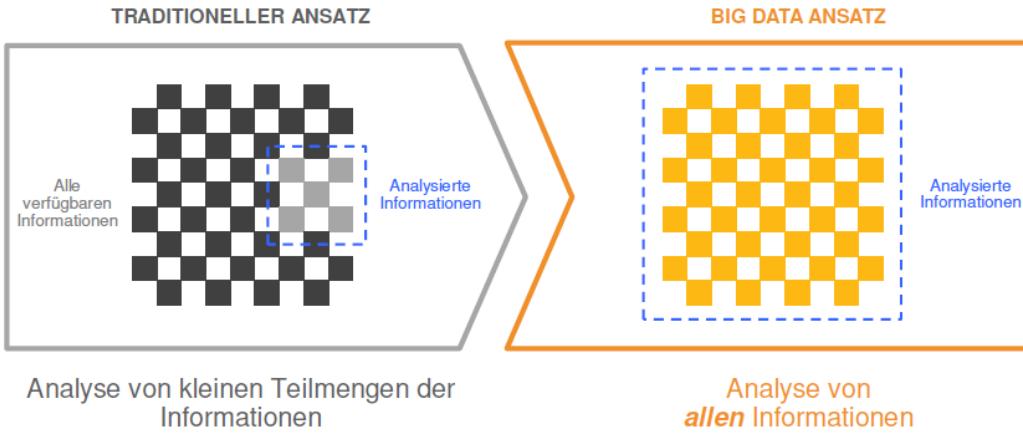
Marketing-(Hype-)Wort

Big im Sinne von Big Oil, Big Business, Big Pharma (=> ein Geschäft, ein Monopol, eine Bedrohung?)
«Letztlich steht der Begriff Big Data jedoch nicht mehr nur für die schiere Menge, Variabilität oder Ausbreitungsgeschwindigkeit von Daten, sondern eher für ein neues Paradigma im Umgang mit ihnen: Daten gelten als verfügbar (zu jedem Thema und aus unterschiedlichsten Perspektiven gesammelt), und es setzt sich die Ansicht durch, dass in deren Auswertung leicht zu schöpfender Wert für ein Unternehmen oder die Gesellschaft als ganzes liegt – „Data“ ermöglicht „Big Gain“. Im Sinne der 5 V's überwältigend „Big“ ausfallende Daten wird es hingegen immer geben.»

Was ist anders mit big data?

- Nutzung von mehr Daten:

z.B. Data Warehouse



© 2016 IBM Corporation

10

Zürcher Fachhochschule

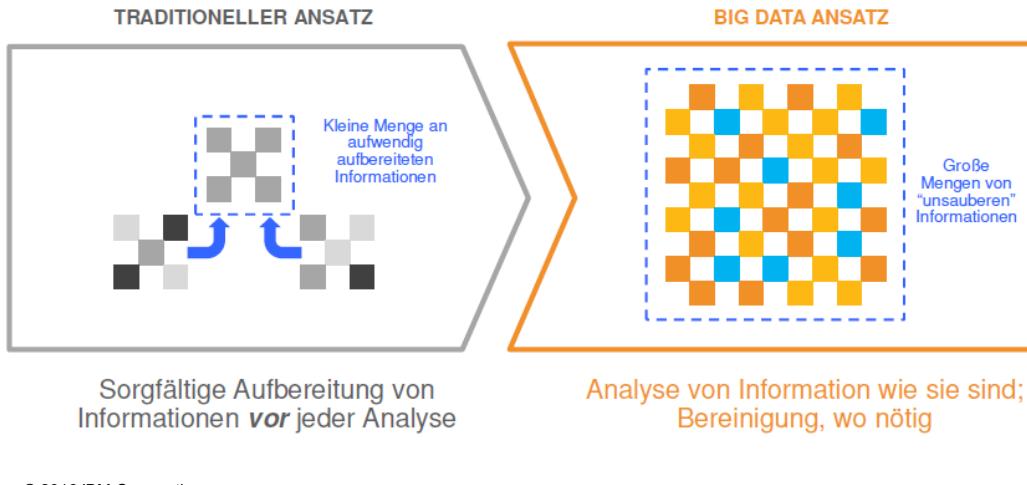
Traditioneller Ansatz: z.B. Data Warehouse

Was ist anders mit big data?

Zürcher Hochschule
für Angewandte Wissenschaften

zhaw School of
Engineering
InIT Institut für angewandte
Informationstechnologie

- Reduzierter Aufwand, um Daten zu nutzen:



© 2016 IBM Corporation

Zürcher Fachhochschule

11

Was ist anders mit big data?

- Daten zeigen den Weg (Korrelationen oft «gut genug») :



Start mit Hypothesen und Test dieser
gegen ausgesuchte Datensätze

Untersuchung **aller** Daten und
die Identifizierung von Korrelationen

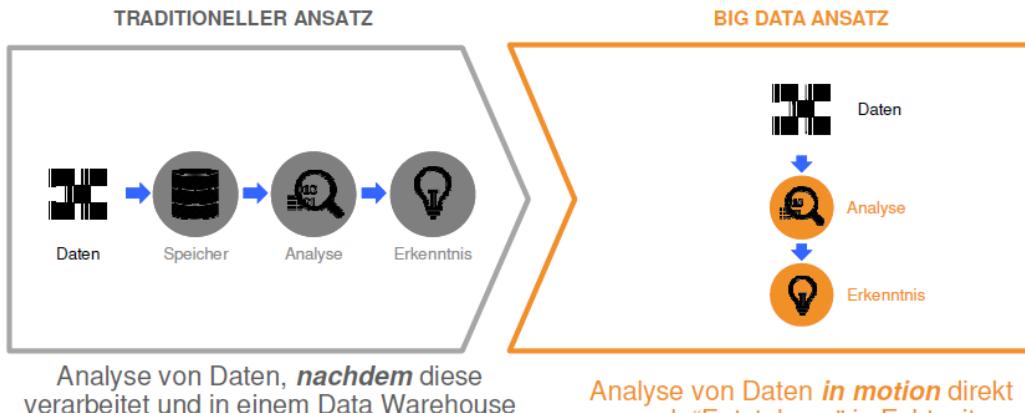
© 2016 IBM Corporation

12

Zürcher Fachhochschule

Was ist anders mit big data?

- Direkte Nutzung von Daten:



© 2016 IBM Corporation

Zürcher Fachhochschule

13

- Standard-Datenbanksysteme
- Non-Standard-Datenbanksysteme
- Skalierung
- NoSQL-Datenbanksysteme

«Standard-Datenbanksysteme»

- Als **Standardanwendungen** bezeichnet man Datenbankanwendungen, die aus dem betriebswirtschaftlichen Umfeld kommen. An ein solches Datenbanksystem werden typischerweise folgende Anforderungen gestellt:
 - Transaktionssicherheit
 - Hohe Performance bei Such- und Änderungsoperationen
 - Verwaltung „einfacher“, stark strukturierter Daten
 - Mächtige deskriptive Abfragesprache: SQL
 - In der Regel ein zentraler DB-Server
- Relationale Datenbanksysteme erfüllen diese Anforderungen in idealer Weise.

15

Zürcher Fachhochschule

- Relationale Datenbanksysteme sind heute in sehr vielen Anwendungsbereichen sehr erfolgreich im Einsatz.
- Entsprechende Produkte wie auch der SQL-Standard werden laufend weiterentwickelt um zunehmend auch neue Anforderungen abdecken zu können:
 - Verarbeitung von XML-Daten
 - "Geografische" Datentypen
 - Unterstützung von OLAP-Anfragen in SQL
 - Unterstützung von grundlegenden Data Mining-Verfahren
 - Verteilung, Parallelisierung (in Grenzen)
 - ...
- Aber es bleiben grundsätzliche Mängel:
 - Fokus auf strukturierte Daten, Skalierung begrenzt
 - Schema-Änderungen sind komplex

- Alle anderen Datenbankanwendungen werden als «**Non-Standard-Anwendungen**» bezeichnet. Diese haben unterschiedlichste Anforderungen an die Datenbankverwaltungssysteme, z.B.:
 - Quasi Echtzeitfähigkeit
 - Verwaltung riesiger Datenmengen (Petabyte)
 - Lange Transaktionen (z.B. im Konstruktionsbereich, CAD/CAM, ...)
 - Abbildung objektorientierter Konstrukte (z.B. Vererbung, Polymorphie) und Speicherung von Objekten. D.h. Speicherung komplexer Daten.
 - Verwaltung von Multimediadaten. Das beinhaltet auch Suchen in Bildern, Videos, ...
 - ...
- Relationale Datenbanksysteme nicht/wenig geeignet, hier werden andere Datenbankverwaltungssysteme gebraucht.

- Nebst den relationalen Datenbankverwaltungssystemen gibt es eine Vielzahl von Systemen, denen ein anderes Datenmodell zugrunde liegt:
 - Netzwerkdatenbanksysteme (veraltet, aber noch im Einsatz, z.B. IDMS)
 - Hierarchische Datenbanksysteme (veraltet, aber noch im Einsatz, z.B. IMS)
 - Objekt-relationale Datenbanksysteme
 - Objektorientierte Datenbanksysteme
 - XML-Datenbanksysteme
 - Temporale Datenbanksysteme
 - Multimediadatenbanksysteme
 - Deduktive Datenbanksysteme
 - Graph-Datenbanksysteme
 - ...

18

Zürcher Fachhochschule

Deduktive Datenbanksysteme versuchen mit einer Deduktionskomponente «Wissen» aus den Daten herzuleiten.

- Im Zusammenhang mit Software versteht man unter **Skalierbarkeit** ein System aus Hard- und Software, dessen Leistung durch das Hinzufügen von Ressourcen oder weiteren Knoten/Rechnern in einem definierten Bereich proportional (bzw. linear) zunimmt.

- **Vertikale Skalierung** (scale up):

- Unter vertikaler Skalierung versteht man ein Steigern der Leistung durch das **Hinzufügen von Ressourcen zu einem Knoten/Rechner** des Systems.
Beispiele dafür wären das Vergrössern von persistentem Speicherplatz, das Hinzufügen einer CPU, oder das Einbauen von mehr Hauptspeicher.
- Charakteristisch für diese Art von Skalierung ist, dass ein System **unabhängig von der Implementierung** der Anwendung schneller gemacht werden kann.

Vertical Scaling

Server	Cost
PowerEdge T110 II (basic) 8 GB, 3.1 Ghz Quad 4T	\$1,350
PowerEdge T110 II (basic) 32 GB, 3.4 Ghz Quad 8T	\$12,103
PowerEdge C2100 192 GB, 2 x 3 Ghz	\$19,960
IBM System x3850 X5 2048 GB, 8 x 2.4 Ghz	\$646,605
Blue Gene/P 14 teraflops, 4096 CPUs	\$1,300,000
K Computer (fastest super computer) 10 petaflops, 705,024 cores, 1,377 TB	\$10,000,000 annual operating cost



PowerEdge R240
Starting at \$589.00



PowerEdge R940xa
Starting at \$9,540.21

Increase performance, ability to scale, and simplify your IT with an entry-level 1U rack server designed for web hosting and multi-purpose applications.

Drive GPU database acceleration for business-critical applications with a powerful, four-socket server. Scale capacity with large internal storage.

Die schnellsten Computer weltweit (Stand: November 2018).^[3]

1. Der aktuelle Spatenreiter ist der Summit des Oak Ridge National Laboratory mit nochmals gesteigerten 143,5 PFLOPS.^[4]
2. Auf den zweiten Platz verbessert sich der Sierra des Lawrence Livermore National Laboratory mit jetzt 94.6 PFLOPS.
3. Den Platz 3 hat der Sunway TaihuLight des chinesischen National Supercomputing Center in Wuxi mit 93.0 PFLOPS.
4. Platz 4 hat der chinesische Tianhe-2 (Milchstraße-2) im National Supercomputing Center in Guangzhou mit 61.4 PFLOPS.
5. Platz 5 hat Piz Daint des Swiss National Supercomputing Centre mit verbesserten 20.23 PFLOPS.
6. Den sechsten Platz nimmt der Trinity des Los Alamos National Laboratory ein mit 20.16 PFLOPS.
7. Auf den siebten Platz fällt AI Bridging Cloud Infrastructure der Universität Tokio mit weiterhin 19.9 PFLOPS.
8. Der SuperMUC NG nimmt Platz 8 weltweit ein mit 19.5 PFLOPS.

21

Zürcher Fachhochschule

FLOP = Floating Point Operations / Second

- Scale up / vertikale Skalierung:

- Vorteile:

- Transparent für DBMS
 - Administrationsaufwand konstant

- Nachteile:

- Hardware-Kosten
 - Lizenz-Kosten DB
 - Skalierung nur in grösseren Stufen möglich → höhere Kosten und ungenutzte Leistung



Quelle: ibm.com

- **Horizontale Skalierung** (scale out):

- Im Gegensatz zur vertikalen Skalierung sind der horizontalen Skalierung keine Grenzen (aus Sicht der Hardware) gesetzt. Horizontale Skalierung bedeutet die Steigerung der Leistung eines Systems **durch das Hinzufügen zusätzlicher Rechner/Knoten**. Die Effizienz dieser Art der Skalierung ist jedoch stark von der Implementierung der Anwendung abhängig, da nicht jede Software gleich gut parallelisierbar ist.
- Charakteristisch für diese Art von Skalierung ist, dass ein System speziell gebaut sein muss um die zusätzlichen Rechner/Knoten auch optimal nutzen zu können.

23

Zürcher Fachhochschule

- Scale out / horizontale Skalierung:

- Vorteile:

- Kostengünstigere Hardware
 - Skalierung in kleineren Stufen möglich

- Nachteile:

- Last- und Datenverteilung notwendig
 - Ggf. verteilte Protokolle (2PC, Replikation)
 - Erhöhte Fehlerrate (mehr und einfache Hardware)
 - Erhöhter Administrationsaufwand



Quelle: eggmusic.com

2PC: Zwei-Phasen-Commit:

Bei verteilten Datenbanken wird ein Commit einer Transaktion zunächst (erste Phase) lokal je Datenbank durchgeführt und anschliessend (zweite Phase) wird der Commit über die verteilten Datenbanken «synchronisiert».

- Unter NoSQL (**Not only** SQL) wird eine neue Generation von Datenbanksystemen verstanden, die meistens einige der nachfolgenden Punkte berücksichtigen:
 - Das zugrundeliegende Datenmodell ist **nicht relational**.
 - Das System ist **schemafrei** oder hat nur schwächere Schemarestriktionen.
 - Das System bietet eine **einfache API**.
 - Die Systeme sind von Anbeginn an auf eine **verteilte und horizontale Skalierbarkeit** ausgerichtet.
 - Aufgrund der verteilten Architektur unterstützt das System eine einfache **Datenreplikation**.
 - Dem System liegt meistens auch ein **anderes Konsistenzmodell** zugrunde: Eventually Consistent und BASE (**Basically Available, Soft State und Eventually Consistent**), aber nicht ACID.
 - Das NoSQL-System ist Open Source (????).
- Vieles in der Entwicklung, sehr dynamisches Gebiet!

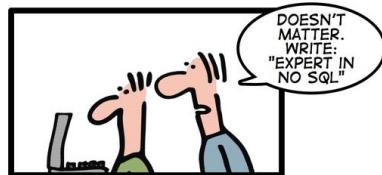
25

Zürcher Fachhochschule

Die Begriffe Eventually Consistent und BASE werden in den nachfolgenden Folien genauer erklärt.

Eventual Consistent: Schreiboperationen werden nicht unmittelbar auf alle Replikas propagiert.
BASE: **B**asically **A**vailable, **S**oft **S**tate und **E**ventually **C**onsistent.

HOW TO WRITE A CV



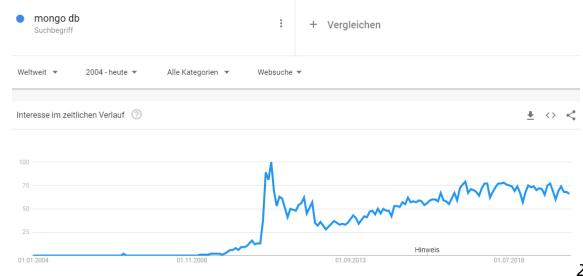
<http://geekandpoke.typepad.com/geekandpoke/2011/01/nosql.html>

Leverage the NoSQL boom

26

NoSQL – Entwicklung

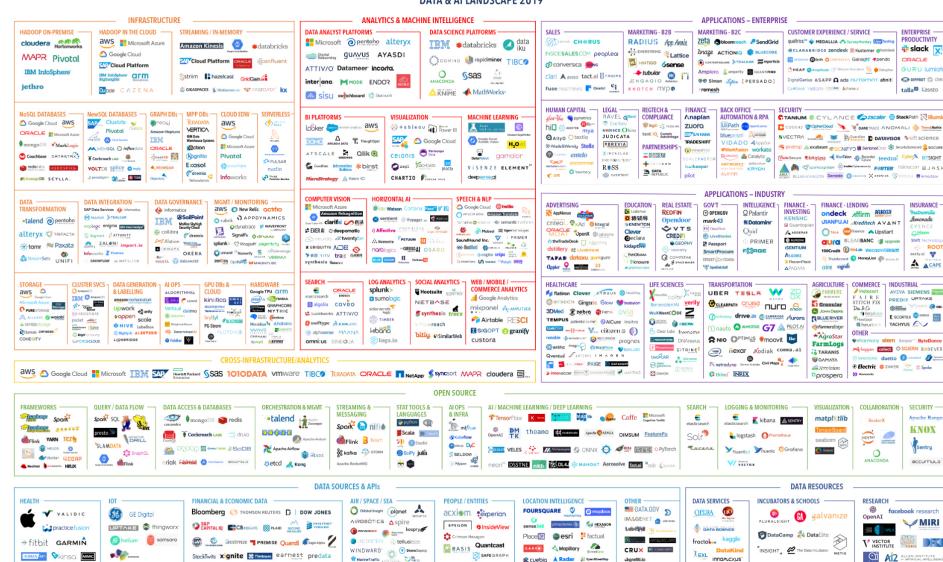
- Einige Meilensteine der Entwicklung von NoSQL:
 - 1979 erste Key/Hash-Datenbank von Ken Thompson.
 - In den 80er-Jahren entstanden Lotus Notes und BerkeleyDB.
 - 1998 tauchte der Begriff NoSQL zum ersten Mal auf. Dabei handelte es sich um eine Datenbank, die zwar relational war, aber kein SQL-Interface hatte.
 - 2000 kam mit dem "Web 2.0" ein Entwicklungsschub.
 - 2004 BigTable-Datenbanksystem von Google. Ähnliche Ansätze bei Yahoo, Amazon, MySpace, Facebook, ...
 - 2006 bis 2009 entstanden die heutigen «klassischen» NoSQL-Systeme wie HBase, CouchDB, MongoDB.
 - Im Mai 2009 tauchte der heutige Begriff NoSQL in einem Weblog von Eric Evans auf.



27

NoSQL – Systeme

Inzwischen gibt es eine Vielzahl von Systemen die unter dem Begriff Big Data / NoSQL laufen:

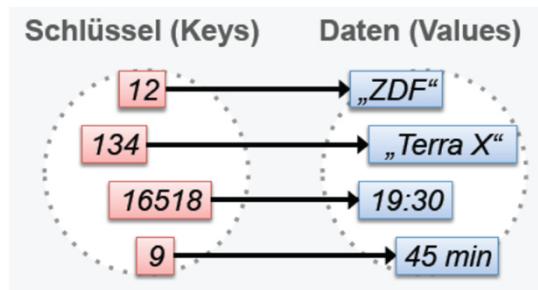


http://mattturck.com/wp-content/uploads/2019/07/2019_Matt_Turck_Big_Data_Landscape_Final_Fullsize.png

28

- Aufgrund der Vielfalt an Konzepten ist eine Klassifizierung schwierig. In Anlehnung an <http://nosql-database.org/> wird oft unterschieden zwischen Core NoSQL und Soft NoSQL-Systemen:
- Core NoSQL Systems (siehe nachfolgende Folien):
 - Key/Value Stores
 - Wide Column Stores
 - Document Stores
 - Graphdatenbanken
 - ...
- Soft NoSQL Systems:
 - Object Datenbanken
 - Grid & Cloud Datenbankanwendungen
 - XML Datenbanken
 - Und viele mehr...

- Eigenschaften:
 - Idee uralt ("hash-tables").
 - Verwendet Schlüssel (Keys) zur Indizierung von Daten (Values).
 - Struktur der Daten ist nicht vorgegeben (schemilos).
 - Datentyp ist (zumeist) nicht vorgegeben.
 - Queries sind auf Schlüssel beschränkt.
 - Typische Grundoperationen:
 - set (key, value)
 - value = get (key)
 - delete (key)
 - Beispiele:
 - DynamoDB
 - Azure Table Storage
 - Berkeley DB
 - Memcache DB
 - Redis
 - ...



30

- Eigenschaften:
 - Ähnlich zu Key Value Stores – jedoch mit mehrdimensionalen Schlüsseln.
 - Performance-Vorteil gegenüber relationalen Datenbanken bei Operationen über viele Einträge mit Schlüsseln geringer Dimensionalität (wenige Spalten). Einsatz z.B. bei Data Warehouses/BI.
 - Besser komprimierbar.
 - Beispiele:
 - Hadoop / HBase
 - Amazon Simple DB
 - MonetDB
 - Cassandra
 - ...

Column Store relationale DB:



31

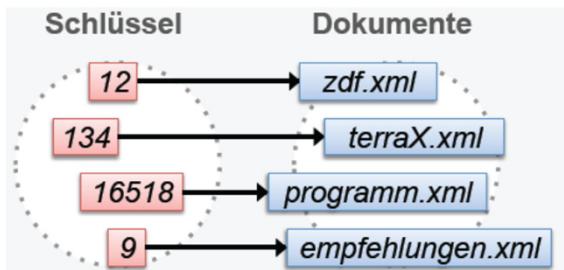
Zürcher Fachhochschule

Achtung: zwei verschiedene Konzepte

Der Begriff spaltenorientierte Datenbank wird unglücklicherweise für zwei verschiedene Arten von Datenbankmanagementsystemen verwendet, die wesentlich mehr Unterschiede als Gemeinsamkeiten haben.

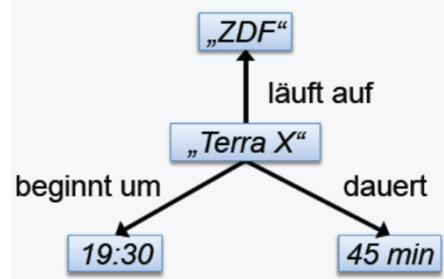
- Einerseits wird der Begriff als deutsche Bezeichnung für Wide Column Stores verwendet (bezieht sich auf den Text auf der Folie). Wide Column Stores, auch Extensible Record Stores und auf deutsch oft spaltenorientierte Datenbank genannt, speichern Daten in Datensätzen mit potentiell sehr vielen dynamischen Spalten ab. Da der Spaltenname ähnlich wie der Schlüssel der Datensätze nicht fix ist, und ein Datensatz auch Milliarden von Spalten haben kann, können Wide Column Stores als zweidimensionale Key-Value Stores angesehen werden.
- Andererseits bezeichnet man damit relationale Datenbankmanagementsysteme, welche die Daten nicht wie üblich zeilenweise, sondern spaltenweise abspeichern (bezieht sich auf das Bild auf der Folie). Dieses nur scheinbar nebensächliche Implementierungsdetail hat zur Folge, dass damit gewisse Arten von Abfragen effizienter abgearbeitet werden können. Das sind im wesentlichen Aggregierungen von Werten über viele Zeilen hinweg, wie sie in Data Warehouse und anderen analytischen Applikationen häufig vorkommen. Auf Englisch wird dieser Typ von DBMS oft als column-oriented DBMS bezeichnet, aber auch dort werden die Begriffe nicht ganz einheitlich getrennt.

- Eigenschaften:
 - Kleinste logische Einheit: «Dokument» identifiziert über DocumentID.
 - Daten sind Dokumente, die nach einem bestimmten Datenformat (XML, JSON, etc.) strukturiert sind.
 - Schemafrei, d.h. Anwendung übernimmt Schema-Verantwortung.
 - Neben «Schlüssel-Anfragen» kann auch nach Inhalten (z.B. XML-Attribut) oder Meta-Daten (z.B. Tags, Sammlungen, etc.) gesucht werden.
 - Anfragesprachen sind abhängig vom Datenformat:
 - XML: XPath, XQuery
 - JSON: JSONiq (<http://jsoniq.org/>)
 - ...
- Beispiele:
 - MongoDB
 - Azure DocumentDB
 - ...



32

- Eigenschaften:
 - Daten = Knoten, Beziehungen der Daten zueinander = Kanten.
 - Daten können über ihre (relative) Position im Graphen identifiziert werden.
 - (traversieren) – es wird kein (globaler) Schlüssel benötigt.
 - Daten können sehr flexibel strukturiert werden.
 - Im Gegensatz zu relationalen Datenbanken sind keine Joins notwendig.
- Beispiele:
 - Neo4J
 - Apache Graph
 - Allegro Graph
 - ...



33

Gewünschte Eigenschaften bei verteilten Datensystemen:

1. Konsistenz (C, Consistency)

Alle Knoten sehen zur selben Zeit dieselben Daten

(diese Konsistenz sollte nicht verwechselt werden mit der Konsistenz der ACID-Transaktionen, welche die innere Konsistenz eines Datenbestandes betrifft).

2. Verfügbarkeit (A, Availability)

Ausgefallene Knoten beeinflussen die akzeptablen Antwortzeiten

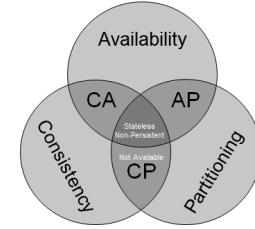
(Verfügbarkeit) der anderen Knoten nicht. Alle Anfragen an das System werden stets beantwortet.

3. Partitionstoleranz (P, Partition Tolerance)

Das System arbeitet auch bei Verlust von Nachrichten, einzelner Netzknoten oder Partitionen des Netzes weiter.

CAP-Theorem (formuliert von Eric Brewer 2000, bewiesen von S. Gilbert & N. Lynch, 2002):

→ Ein **verteiltes System** kann jeweils nur zwei der drei Eigenschaften gleichzeitig erfüllen, jedoch nicht alle drei.



- Anwendungen des CAP-Theorems:
 - Für klassische (verteilte) Datenbankanwendungen ist vor allem C wichtig; ggf. müssen Einschränkungen bei A oder P akzeptiert werden.
 - Bestimmte (Web-)Anwendungen fokussieren auf A und P und können ggf. Einschränkungen bei C akzeptieren.

- Beispiele:

- AP – Domain Name System (DNS):

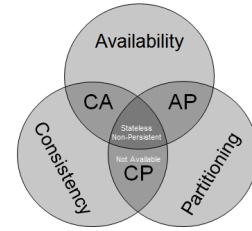
Das DNS fällt in die Kategorie AP. Die Verfügbarkeit ist extrem hoch, ebenso Toleranz gegenüber dem Ausfall einzelner DNS-Server. Allerdings ist die Konsistenz nicht immer sofort gegeben: es kann mitunter Tage dauern, bis ein geänderter DNS-Eintrag an die gesamte DNS-Hierarchie propagiert ist und damit von allen Clients gesehen wird.

- CA – Relationales Datenbank Management System (RDBMS):

«Standard-Anwendungen» fallen in der Regel in die Klasse CA. Z.B. das im Kontext der Transaktionsverwaltung diskutierte Kontoverwaltungsbeispiel.

- CP – Geldautomat:

Für Geldautomaten ist die Konsistenz der Daten oberstes Gebot. Diese Anforderung muss auch bei Störungen im Datenverkehr sichergestellt werden (Partitionstoleranz). Gegenüber der Konsistenz und der Partitionstoleranz ist die Verfügbarkeit zweitrangig (CP): bei Netzwerkstörungen soll ein Geldautomat oder ein anderer Service eher nicht verfügbar sein als nicht korrekte Transaktionen abwickeln.



Akzeptanz von **BASE** (auch von Eric Brewer)
(Basically Available, Soft State, Eventually Consistent)

- Konsistenz (nicht die Konsistenz gemäss ACID) wird erst nach einem Zeitfenster der Inkonsistenz erreicht und nicht unmittelbar nach jeder Transaktion.
- Zeitfenster z.B. definierbar basierend auf Anzahl replizierender Knoten, durchschnittlichen Reaktionszeiten, durchschnittlicher Last o.ä.
- Weitere Spezifikationen dieses «Konsistenzbegriffes» aktuell in der Entwicklung, u.a. read-your-writes consistency, monotonicreadconsistency etc.

37

Zürcher Fachhochschule

BASE ist ein Begriff, der als Gegenstück zu den strengen ACID Kriterien geschaffen wurde, um das in NoSQL Systemen typische etwas flexiblere Verständnis von Konsistenz zu charakterisieren.

BASE steht für:

- Basically Available
- Soft state
- Eventual Consistency

Selbst der Erfinder dieses Akronyms, Eric Brewer, gibt zu, dass die Definition etwas gekünstelt ist, um einen sprachlich einprägsamen Gegensatz zwischen BASE (eng. für Lauge) zu ACID (eng. für Säure) zu schaffen. Es gibt daher auch keine präzise Definition für "basically available" und "soft state". Jedenfalls steht BASE für ein Designprinzip, welches absolute Konsistenz aufgibt, stattdessen die Verfügbarkeit des Systems erhöht, und dadurch zwischenzeitlich in einem etwas undefinierten Zustand sein kann.

Wesentlich störender als die sprachlich holprige Konstruktion ist jedoch, dass Consistency in BASE und in ACID für verschiedene Konzepte steht, nämlich einerseits für Konsistenz in verteilten Systemen, andererseits für das was wir als Integrität im Zusammenhang mit Transaktionen beschreiben. Somit könnte man sagen, dass BASE und ACID zwei Seiten einer Medaille sind, aber nicht derselben Medaille. (<https://db-engines.com/de/article/BASE>)

- Vorteile:

- Flexible und kostengünstige horizontale Skalierung (scale out).
- Verarbeitung riesiger Datenmengen mit kostengünstiger Software.
- Hochgradig parallelisierbare Anfrageverarbeitung mit MapReduce.
- Schemaflexibilität (falls benötigt).
- ...

- Nachteile:

- Ggf. Abstriche bei Konsistenz.
- Erhöhter Aufwand für Entwicklung.
- Proprietäre, weniger mächtige APIs / „Anfragesprachen“.
- Bisher kaum Tools für Performance-Analyse und DB-Administration.
- Noch relativ wenig Erfahrungen vorhanden.
- ...

38

Zürcher Fachhochschule

MapReduce (vereinfacht):

MapReduce ist Modell, in welchem Daten in zwei Phasen (map und reduce), auf (sehr) viele Verarbeitungsknoten verteilt, verarbeitet werden. In der ersten Phase (map) arbeiten die Prozesse isoliert auf ihrem Teil der Daten und erzeugen eine Ausgabe (intermediate file). In der zweiten Phase (reduce) werden diese Zwischenresultate wiederum durch parallel laufende Prozesse zum Resultat verdichtet.

Beispiel MapReduce: In einem Key/Value-Store, dessen Values alle Worte sind, soll festgestellt werden, wie viele Anagramme vorkommen. Die Daten des Key/Value-Store seien z.B. auf 5 Knoten verteilt. Ein solcher Key/Value-Store eines einzelnen Knotens könnte z.B. wie folgt aussehen: «1/Fehler; 2/Nix; 3/Helper»

- Map: Im ersten Schritt bildet jeder Knoten für sich isoliert jedes Key/Value-Paar in ein neues Key/Value-Paar ab (hier werden die Buchstaben sortiert). Für unser Beispiel: «EEFHLR/Fehler; INX/Nix; EEFHLR/Helper». Im Beispiel erhalten wir für die 5 Knoten 5 Zwischenresultate.
- Reduce: Im 2. Schritt werden diese Resultate zusammengeführt. Soll dies wieder verteilt auf 5 Knoten erfolgen, müssen diese Zwischenergebnisse zunächst sinnvoll verteilt (Partitioning-Function) und gebündelt (Combiner-Function) werden. So, dass z.B. alle Keys von «A» bis «N» auf dem ersten Knoten zu liegen kommen. Im Reduce würden die Resultate nun sinnvoll gebündelt, z.B. «EEFHLR:Fehler;EEFHLR:Helper;INX:Nix...» (wir haben ein Anagramm gefunden).

- Sehr dynamische Entwicklung
- Viele Technologien interessant, aber noch etwas «unreif».
- Die Unterstützung im Bereich der Abfragesprachen muss noch stark verbessert werden.
- Trend: Erweiterung bestehender relationaler DBMS und Data Warehouse Systeme:
 - MapReduce (Hadoop) Integration
 - Spaltenindizes (Columnstore-Index)
 - ...
 - Beispiele SQL Server:
 - 2019 Hadoop, Apache Spark, Java, Scala, Big Data Clusters
 - 2017 SQL Graph Architecture
 - 2016 Python, R
 - 2014 In-Memory OLTP
 - 2012 Column Stores

39

Zürcher Fachhochschule

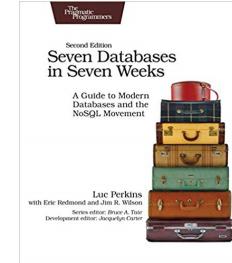
Hadoop ist ein freies, in Java geschriebenes Framework für skalierbare, verteilte arbeitende Software (gepflegt durch Apache Software Foundation). Es basiert auf dem MapReduce-Algorithmus.

Literaturhinweise

Zürcher Hochschule
für Angewandte Wissenschaften



- Luc Perkins, E. Redmond & J. Wilson:
Seven Databases in Seven Weeks
Pragmatic Bookshelf, 2018
ISBN 978-1680502534



- Martin Kleppmann:
Designing Data-Intensive Applications
O'Reilly Media, 2017
ISBN 978-1449373320



Zürcher Fachhochschule

40

Ausblick

Zürcher Hochschule
für Angewandte Wissenschaften



- Das nächste Mal: Abschluss, Fragen
- Lesen: Nichts!

41

Zürcher Fachhochschule