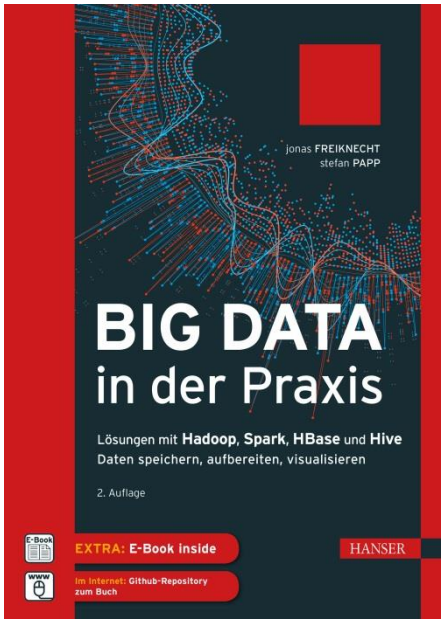
HANSER



Leseprobe

zu

Big Data in der Praxis

Jonas Freiknecht
Stefan Papp

ISBN (Buch): 978-3-446-45396-8
ISBN (E-Book): 978-3-446-45601-3

Weitere Informationen und Bestellungen unter
http://www.hanser-fachbuch.de/
sowie im Buchhandel

Inhalt

2.3 Eingliederung in BI und Data Mining

Um die Begriffe BI und Data Mining in Relation zu Big Data setzen zu können, gilt es, diese im Vorfeld zu definieren. Kemper, Mehanna & Unger bezeichnen BI als Filter, der Daten in strukturierte Information umwandle (Kemper, er al., 2010). Gartner hingegen konstatiert etwas ausführlicher, dass BI ein Überbegriff für Anwendungen, Infrastruktur, Werkzeuge und Best Practices sei, die den Zugriff auf und die Analyse von Informationen ermöglichen, um Entscheidungsfindung und Performance zu erhöhen (Gartner). Hält man sich nun strikt an die Definitionen, besteht der Unterschied zwischen BI und Big Data darin, dass BI sich auf bereits vorliegende Informationen bezieht, die dazu noch strukturiert sind und sich auf einen eindeutigen Kontext beziehen.

Das Ziel von BI und der Big-Data-Explorationen ist jedoch dasselbe, nämlich aus vorhandenen Daten neue Erkenntnisse zu gewinnen, die der Entscheidungsfindung bei vorher definierten Fragestellungen dienen. Der Trend bei Big Data geht dabei auch oft in die Richtung zu lernen, was man noch nicht weiß.

BI ist jedoch mittlerweile mehr als diese einfache Begriffsdefinition. Es hat sich in den letzten Jahren zu einem festen Prozess samt einem Set aus technischen Werkzeugen entwickelt, um das Berichtswesen in Unternehmen zu automatisieren. Dazu gehören die Datenaufbereitung, die Datenspeicherung in DWHs sowie deren Darstellung aus verschiedenen Perspektiven.

Welche Techniken, Methoden und Arbeitsschritte werden aber nun angewandt, um Informationen aus vorliegenden Daten zu extrahieren? Die Antwort darauf gibt der sogenannte KDD-Prozess (*Knowledge Discovery in Databases*).

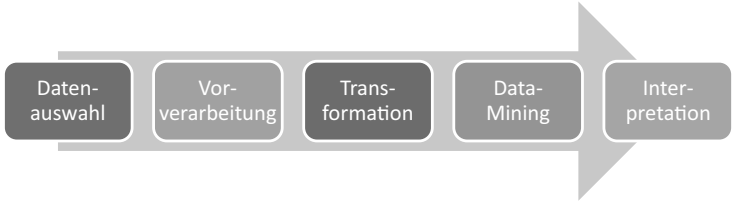


Bild 2.5 Der KDD-Prozess nach (Kononenko, et al., 2007)

Der (iterative und interaktive) KDD-Prozess hat das Ziel, gültige, neue, nützliche und verständliche Muster in vorhandenen Daten zu erkennen (Fayyad, et al., 1996). Wirft man nun einen Blick auf den vierten Schritt des in Bild 2.5 illustrierten Ablaufs, so ist zu erkennen, dass Data Mining einen Teil des KDD-Prozesses darstellt. Dieser nimmt gesäuberte, korrigierte und transformierte Daten entgegen, extrahiert daraus Muster und Zusammenhänge und gibt diese zur Interpretation frei. Quellen müssen, anders als der Begriff KDD vermuten lässt, nicht zwingend Datenbanken sein, sondern können auch als simple Datensätze gesehen werden, z. B. als Flat Files, CSV, XML oder Dateisystemstrukturen. Wichtig ist, dass diese bereits im Vorfeld aufbereitet wurden. Zu dieser Aufbereitung (*Preprocessing*) gehören:

- Formatanpassungen (z. B. Datums- und Zahlenformate)
- Korrigieren von Ausreißern (Messfehler, Verarbeitungsfehler, bewusste Falschangabe)
- Auffüllen dünn besetzter Daten

HINWEIS: In späteren Kapiteln wird der KDD-Prozess im Detail durchgegangen.

Stellt man nun die drei Begriffserklärungen BI, Data Mining und Big Data einander gegenüber, so erkennt man schnell einige Gemeinsamkeiten sowie Unterschiede.

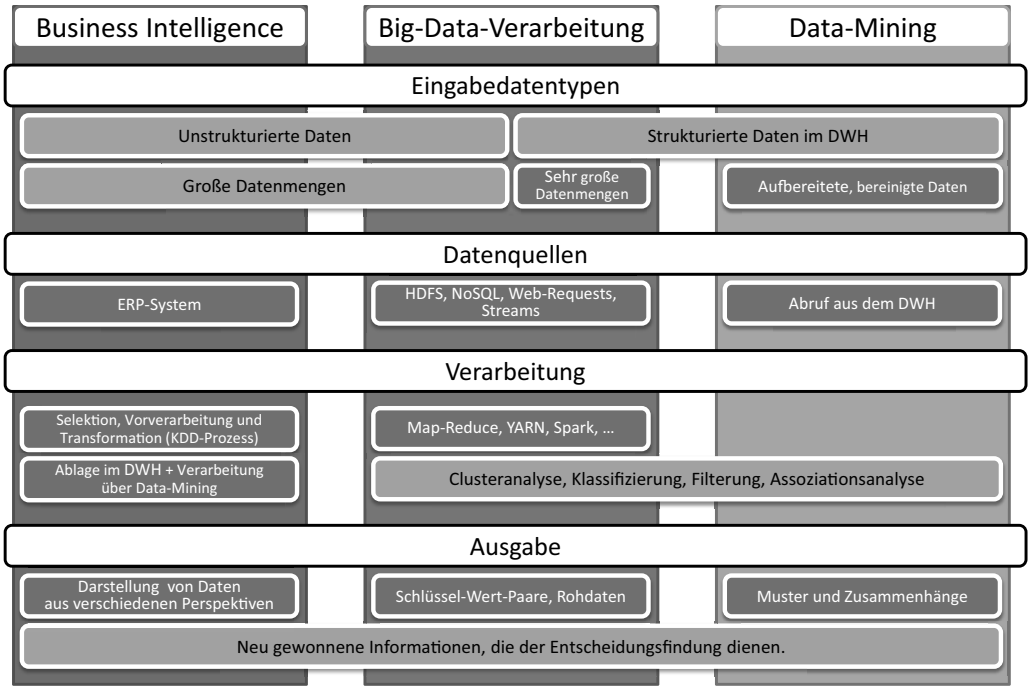


Bild 2.6 Definitionsvergleich von BI, Big Data und Data-Mining

Vor Big Data wurden größere Analysen fast ausschließlich in Datenbanken ausgeführt. Es gab dazu auch eine Königsklasse, das MPP (Massive Parallel Processing), das eine parallele Verarbeitung zuließ. Im Kern folgte aber jedes Data Warehouse den Prinzipien der relationalen Datenbanken.

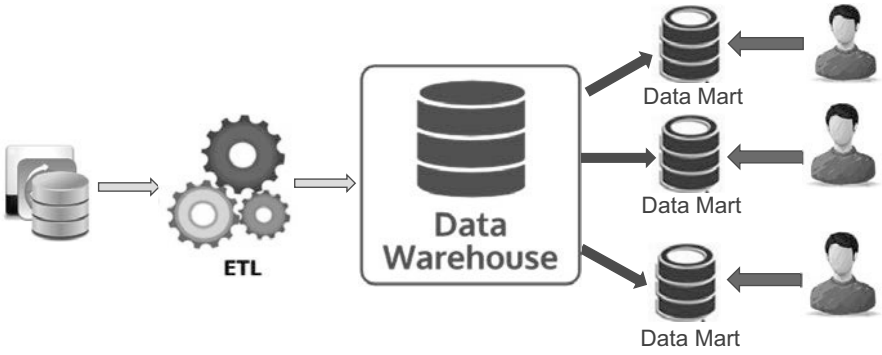


Bild 2.7 Das Data Warehouse

Das obige Beispiel illustriert den Prozess in den DWHs. Daten werden aus ihren Quellen entnommen, in sogenannten ETL-Tools aufbereitet und als relationale, strukturierte Daten im DWH abgelegt. Vom DWH werden Extrakte in Data Marts abgelegt, auf die dann Analysten zugreifen, um Ergebnisse abzuleiten.

In der ersten Evolutionsstufe von Hadoop empfahlen manche DWH-Systemhersteller, Hadoop als billigen Speicherplatz zu nehmen. Hadoop würde dann die Rolle von ETL übernehmen, die Daten aufbereiten und ins DWH laden. Die Auswertung erfolgte dann im DWH. Für Spezialfälle boten die Systemhersteller auch Möglichkeiten an, sogenannte Kaltdaten aus Kostengründen auf Hadoop zu belassen und sie bei Bedarf ins DWH zu laden.

Der Vorteil vom DWH wurde so angepriesen, dass bekannte analytische Verarbeitungsmethoden (Stichwort *Online Analytical Processing Cube*) und eine reichhaltige Funktionsbibliothek zur Verfügung stehen. Auch steckte jahrelanges Know-how in Komponenten wie der SQL Engine des DWHs. SQL auf Hadoop war zumindest am Anfang extrem langsam und hatte bei Joins eine schlechte Performance.

Auch die Rolle von sogenannten Self-Service BI Tools (wie Tableau) und vor allem Excel darf nicht unterschätzt werden. Soll also in einer Grafik der Gewinn eines Unternehmens inklusive aller Tochterfirmen angezeigt werden, müssen im Reporting-Werkzeug lediglich vom Benutzer die bereits ermittelten Gewinne der Tochterfirmen dem Gesamtgewinn hinzuaddiert werden. Um das zu bewerkstelligen, gibt es Konnektoren zu verschiedenen Datenlieferanten.

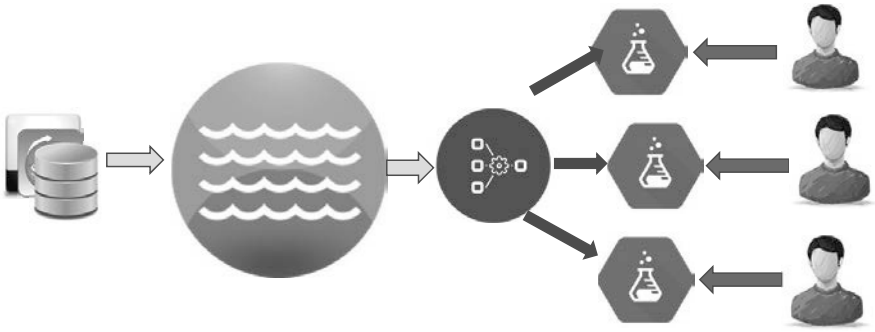


Bild 2.8 Der Data Lake

Mit Hadoop hat sich vieles geändert. Daten werden sofort auf den Hadoop-Cluster geladen. Wir sprechen hier auch vom Data Lake. Man sprach von ETL (Extract Transform Load), wenn die Daten vor der Beladung vom DWH noch transformiert wurden, bevor sie abgelegt werden konnten,

Im Falle des Data Lakes spricht man aber von ELT (Extract Load Transform). Daten werden aus den Quellsystemen gezogen, auf den Data Lake geladen und schließlich in ein Zielsystem transformiert.

Bei der Aufbereitung von Daten werden die unstrukturierten Daten oft in ein strukturiertes Schema überführt. Daten, die in Apache Hive-Tabellen abgelegt werden, können auch als Datenlieferanten von Self-Service BI Tools wie Tableau dienen.

Mit der Zeit wurde Hadoop zu einer Datenplattform, die auf einem verteilten Dateisystem aufbaut und zahlreiche Möglichkeiten bietet, die Daten aufzubereiten. Diese Daten können in Strukturen gebracht und gleichzeitig analysiert werden.

Die Reise war damit aber noch nicht zu Ende. Das komplette Open-Source-Ecosystem von Big Data bietet mittlerweile zahlreiche Werkzeuge an, die die Verarbeitung von Daten noch weiter vereinfachen und neue Methoden wie die Echtzeitverarbeitung mit sich bringen.

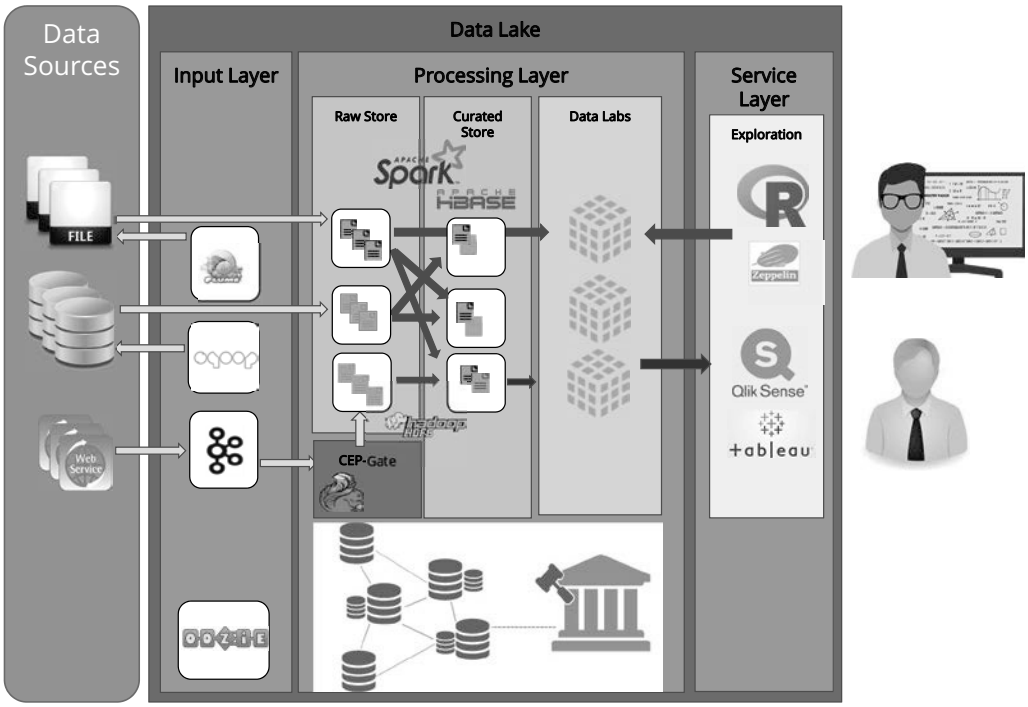


Bild 2.9 Big-Data-Referenzarchitektur

Bild 2.9 zeigt eine komplette Referenzarchitektur mit Big-Data-Komponenten. Es landen unterschiedliche Daten auf einer Datenplattform (in diesem Beispiel Hadoop). Sind die Daten auf Hadoop geladen, werden sie für verschiedene Szenarien unterschiedlich weiterverarbeitet und dann in Tabellen abgelegt, auf die dann analytische Werkzeuge zugreifen können.

Architekturen wie die Lambda- und Kappa-Architektur zeigen, dass man auch Architekturmuster bauen kann, die auf Echtzeitverarbeitung aufbauen.

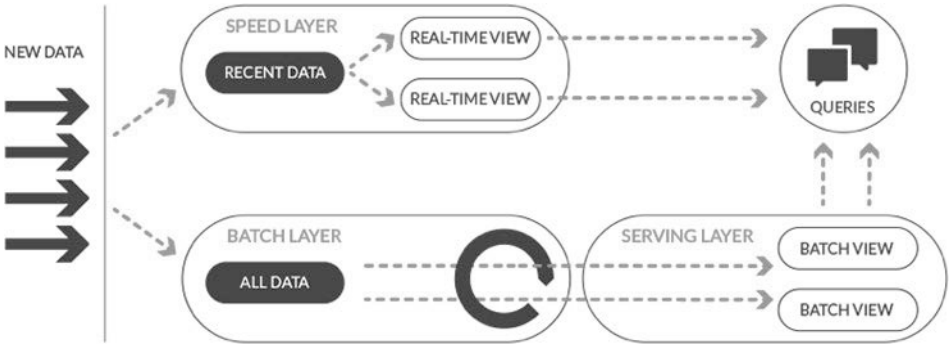


Bild 2.10 Big-Data-Architekturen

Zusammengefasst kann man sagen, dass in in Lambda-Architekturen Daten beim Laden in zwei Bereiche geteilt werden. Im Speed Layer werden die Daten im Regelfall in Memory gehalten. Im Batch Layer werden alle Daten archiviert. Verschiedene Prozesse verarbeiten die Daten kontinuierlich und bereiten die Ergebnisse im Serving Layer für einen schnellen Zugriff auf. Details können im Buch von Nathan Marz nachgelesen werden (Marz, 2015).

Ein weiteres zentrales Thema, das im Umfeld von Big Data anzutreffen ist, ist Data Governance. Dieser Begriff umfasst die Verwaltung von Daten, konkret geht es um Katalogisierung, Klassifikation und weitere Prozesse wie z. B. Retention, um Daten nach einem bestimmten Ablauf auch wieder zu löschen. In Data Governance spielen auch Themen wie Sicherheit und Datenqualität hinein.

Ein weiteres Thema sind natürlich alle Methoden, um Daten zu analysieren. Nicht umsonst hat der Beruf des Data Scientist einen hohen Stellenwert bekommen. Auch gibt es mittlerweile zahlreiche Kurse zu Deep Learning und Machine Learning.

Als dieses Buch in der ersten Version geschrieben wurde, wurden fast ausschließlich Hadoop-Technologien beschrieben. In einem Interview erklärte Ted Dunning, Chief Architect von MapR, dass Open-Source-Technologien deswegen so erfolgreich sind, weil sie über saubere Schnittstellen verfügen. Einzelne Komponenten wären somit leichter austauschbar.

Die Konsequenz daraus ist, dass viele Open-Source-Big-Data-Systemlandschaften wie ein Baukasten aufgebaut sind, wo einzelne Komponenten austauschbar sind. Das trifft auch auf Hadoop zu. MapR hat beispielsweise ein eigenes Dateisystem geschrieben, das auf der HDFS-API aufbaut und mehr Funktionalität aufweist als HDFS. Neue Komponenten wie Apache Spark nutzen nur mehr Schnittstellen und können Daten auf HDFS schreiben, aber auch auf viele andere Plattformen wie Cassandra oder Amazon S3. Die zentrale Botschaft ist, dass Hadoop in diesem Buch zwar nach wie vor einen hohen Stellenwert hat, aber dieses Buch kein reines Hadoop-Buch ist, sondern viele Technologien vorstellt, die man mit Hadoop verwenden kann, die aber im eigentlichen Sinne von Hadoop unabhängig sind.

- *Geografische Daten* werden oft als Karte, z. B. in Bild 7.2 als *Choroplethenkarte*, dargestellt, um bestimmte Ausprägungen eines Merkmals auf ein bestimmtes Areal abzubilden oder die Aufmerksamkeit des Betrachters durch Marker auf bestimmte Koordinaten, Städte, Flüsse oder Strecken zu lenken.

Kurniawan behält sich vor, dass es noch andere Datentypen gibt, die in keines dieser Schemata passen. Hier wäre beispielsweise der Datensatz anzuführen, der einer *Tag Cloud*, dem Diagramm der *Web-2.0-Ära*, zugrunde liegt. Dabei handelt es sich um eine Liste nicht zueinander in Beziehung stehender Daten, die letztendlich in einer zusammengesetzten, ineinander verwinkelten Textabbildung zusammengeführt werden.

7.3 Visualisieren von Big Data erfordert ein Umdenken

Was wir bisher bezüglich der Datenvisualisierung besprochen haben, ist nicht neu. Die Datentypen, die dargestellt werden, waren auch schon vor Auftreten der Big-Data-Thematik bekannt, und die Tatsache, dass Diagramme unserem Verstand schneller zugänglich sind als bloße Daten, ist auch keine bahnbrechende Erkenntnis. Wo also liegen die Besonderheiten und die Herausforderung bei der Darstellung von Big Data? Klar ist, dass der Datenumfang erst einmal eingeschränkt werden muss. Stellen Sie sich vor, dass Sie Millionenstädte auf einer Landkarte anzeigen wollen und zu jeder Stadt im Sichtfeld Informationen anbieten möchten, z. B. Einwohnerzahl, Gründungsjahr oder das Durchschnittseinkommen. Insbesondere sollen des Weiteren die Städte hervorgehoben werden, die über drei Millionen Einwohner aufweisen. Aus diesem einfachen Szenario ergeben sich im Grunde zwei grob formulierte Anforderungen an die Big-Data-Visualisierung. Diese lauten:

- *Wie können wir viele Daten auf einmal darstellen?*
- *Wie können wir in großen Datenmengen die Aufmerksamkeit auf einen Datensatz lenken?*

Bereits in Abschnitt 3.18.2 wurde erwähnt, dass das Ausdünnen der Daten eine elementare Rolle bei der Aufbereitung von Big Data spielen kann. Dieser Schritt kommt bei der Datenvisualisierung besonders zum Tragen, bedenkt man, dass Diagramme häufig auf Client-Seite via JavaScript oder Flash gerendert werden und die Daten dazu von unserem Server auf den Rechner des Betrachters transportiert werden. Hier müssen wir aus Zeit- und Performancegründen die Datenmenge auf ein akzeptables Minimum reduzieren, um die Ladezeiten kurz und die *Usability* der Anwendung hoch zu halten.

Eine zweite Herausforderung, die sich aus der ersten Frage ergibt, ist die des Platzes, der uns für die Darstellung der Daten zu Verfügung steht. Wie können wir diesen effizient nutzen, sodass der Betrachter trotz der erhöhten Anzahl der Datensätze trotzdem die Aussage des Diagramms erkennt, es thematisch einordnen kann und das Diagramm Interesse weckt und nicht schon im Vorfeld zu komplex wirkt? Und wie heben wir bestimmte Areale hervor, die für den Betrachter besonders interessant sein könnten? Lassen Sie uns die folgenden Abschnitte als möglichen Lösungsansatz für die angesprochenen Punkte betrachten.

7.3.1 Aufmerksamkeit lenken

Jenifer Tidwell (Tidwell, 2011) stellt die Behauptung auf, dass visuelle Werkzeuge bereits Informationen liefern, ohne dass der Betrachter ihnen überhaupt aktiv Beachtung schenkt. Ist diese Behauptung auch noch zu vertreten, wenn Big Data dargestellt wird? Schließlich müssen hier ja viel mehr Daten auf demselben Platz untergebracht werden, und da liegt die Vermutung nahe, dass diese intuitive Informationsaufnahme vielleicht nicht mehr ganz so selbstverständlich erfolgt. Wie also lassen sich wichtige Informationen in Diagrammen auch bei großem Umfang weiterhin hervorheben, sodass diese auf einen Blick ersichtlich sind? Und wie wird vermieden, dass das Hindeuten auf diese Merkmale nicht abermals zu viel Platz auf der Abbildungsfläche in Anspruch nimmt? Zwar ließen sich leicht große Pfeile auf interessante Punkte eines Diagramms zeichnen und beschriften, jedoch ginge dadurch ein wesentlicher Teil der Grafik verloren, der zur Darstellung weiterer Datensätze genutzt werden könnte. Tidwell führt für diesen Fall folgende Möglichkeiten (Bild 7.5) an, um einzelne Bereiche hervorzuheben. Dabei verzichtet sie weitestgehend auf Beschriftungen oder weitere Elemente auf der Grafik, sondern modifiziert die jeweiligen dargestellten Daten nur leicht in Form, Position und Farbe.

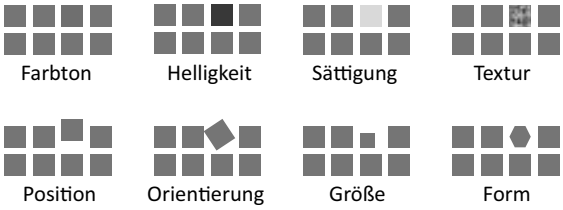


Bild 7.5 Werkzeuge, um Aufmerksamkeit gezielt zu lenken (Tidwell, 2011)

Zu Beginn werden drei Möglichkeiten angeführt, die die Farbe des hervorzuhebenden Datums ändern, genauer: Farbton, Helligkeit und Sättigung. Gewiss könnte man die drei zu einem einzelnen Punkt zusammenfassen, jedoch ergeben sie einzeln genommen verschiedene Möglichkeiten der Darstellung, die bei einer einfachen Änderung des RGB-Wertes (Rot, Grün, Blau) verwehrt blieben. Abermals soll die Choroplethenkarte aus Bild 7.2 als Beispiel dienen, auf der verschiedene Niveaus durch die Helligkeit der Farbe dargestellt werden. Anders als bei einer eigens erdachten Farbskala (z. B. einer *Heat Map*, die Werte von Grün über Gelb nach Rot darstellt), kann so ein klarer Verlauf gezeigt werden, der beispielsweise von 0 bis 100 reicht und keine sichtbaren Schwellenwerte aufweist. Bei der als Beispiel angeführten *Heat Map* ruft jeder Farbwechsel beim Betrachter eine sofortige Assoziation mit einer Zustandsänderung hervor. Die Farbe Grün wird sehr positiv wahrgenommen, Gelb gilt als kritisch und Rot als negativ. Eine Karte, die mit einer einzigen Farbe arbeitet und lediglich deren Helligkeit verändert, führt nicht zu derartigen Interpretationen.

Bezüglich des Farbschemas eines Diagramms gilt es, eine gewisse Konsistenz zu wahren und Farben zu wählen, die dem Betrachter gegenüber angenehm erscheinen. Solche Schemata lassen sich leicht über eigens dafür erstellte Web-Dienste[2] erstellen. Neben der Farbgebung spielt ebenso die Texturierung eine Rolle, die durch ein Muster Datensätze hervorheben kann.

[2] http://colorschemedesigner.com/

Die form- oder positionsändernden Merkmale wie Positionierung, Orientierung, Größe oder Form sind mit Vorsicht zu genießen, da diese voraussetzen, dass alle anderen Daten geordnet und gleich verteilt zueinander ausgerichtet sind und über ein einheitliches Erscheinungsbild verfügen. Weniger aussagekräftig wäre diese Darstellung etwa auf einer Landkarte, da die Grenzen dem Betrachter zwar geläufig sein können, die Umrisse aber so stark variieren, dass bei geringen Änderungen der Form eines bestimmten Areals nicht wahrgenommen werden kann, ob sich dieses nun verändert hat oder nicht.

Bild 7.6 Geburten werden über Formveränderungen auf einer Karte der USA dargestellt (Allen).

So zeigt Bild 7.6 eine Karte, die die Geburtenrate in den USA visualisieren soll. Auch mit Kenntnissen der Topografie von Nordamerika ist es schwer, Aussagen darüber zu treffen, in welchen Staaten die Rate nun besonders hoch ist. Kalifornien, New York, Florida und Texas scheinen etwas ausgeprägter zu sein als die anderen, jedoch ist eine klare Aussage, ohne Beachtung der Farbhelligkeit, nicht zu treffen. Nichtsdestotrotz hat Shawn Allen mit dieser Karte eine fantastische Visualisierungsmethode geschaffen, die ihre ganze Aussagekraft erst entfaltet, wenn der Benutzer aktiv mit dem Diagramm interagiert. Dabei kann die Karte in einer Animation in ihren Normalzustand transformiert werden, wodurch die Ausprägung des Merkmals *Geburtenrate* erst sichtbar wird. Besonders eindrucksvoll ist weiterhin die Möglichkeit, zwischen verschiedenen Merkmalen zu wechseln (Bild 7.7), z. B. mit der Migrationsrate oder der Bevölkerungsdichte. Eine Karte visualisiert hier also durch Nutzung von interaktiver Veränderung von Form, Farbe und Größe 16 Merkmale auf engstem Raum. Das Stichwort Interaktivität soll nun direkt in den nächsten Abschnitt überleiten.

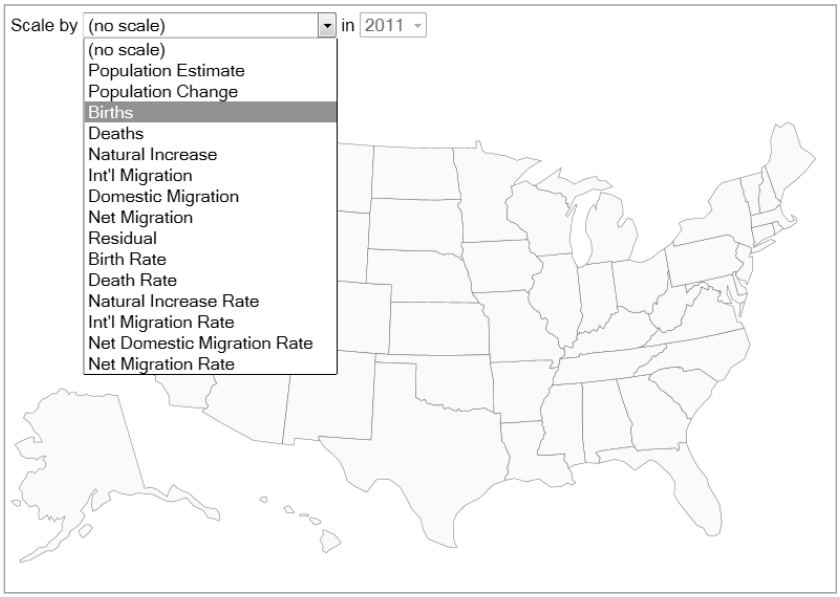


Bild 7.7 Nordamerika ohne farbliche oder strukturelle Veränderung (Allen)

So etwa wird Alaska unten links in Bild 7.6 durch eine geringe Geburtenrate stark komprimiert. Ebenso ergeht es Montana und North Dakota im Norden. Die Karte, die Allen hier entworfen hat, zeigt ihre ganze Aussagekraft erst dadurch, dass der Benutzer verschiedene Kennzahlen aus einer Drop-down-Box auswählt und sich die Abbildung dann sofort entsprechend verformt und einfärbt.

> *„A good visualization is not just a static picture or a three-dimensional (3D) virtual environment that we can walk through and inspect like a museum full of statues. A good visualization is something that allows us to drill down and find more data about anything that seems important.“* – Colin Ware (2012, S. 345)

Wenn nicht alle Kennzahlen auf eine Grafik passen und die Möglichkeit der Interaktion gegeben ist, dann kann dem Benutzer also gestattet werden, die Kennzahlen und somit die Grafik zu manipulieren. Dieses Feature bleibt Print- und anderen statischen Medien verwehrt, bringt jedoch im Bereich der digitalen Visualisierung viele Vorteile mit sich, wie z. B. die exemplarisch gezeigte Änderung des der Visualisierung zugrunde liegenden Datensatzes.

7.3.2 Kontextsensitive Diagramme

Die grafische Darstellung großer Datenmengen benötigt Platz. Die Betrachtung in *Microsoft Excel* oder *PowerPoint* erstellter Diagramme zeigt, dass ein beachtlicher Teil für Beschriftungen, Legenden und Indizes verwendet wird. Häufig lassen sich diese jedoch vermeiden, ohne die Aussagekraft eines Diagramms zu reduzieren. Besonders durch den Einsatz von kontextsensitiven Diagrammen, die durch ihr bloßes Aussehen bereits eine Thematik kom-

munizieren, kann ein signifikanter Teil der Beschriftungen eingespart werden. Diese These soll nun durch ein passendes Szenario untermauert werden.

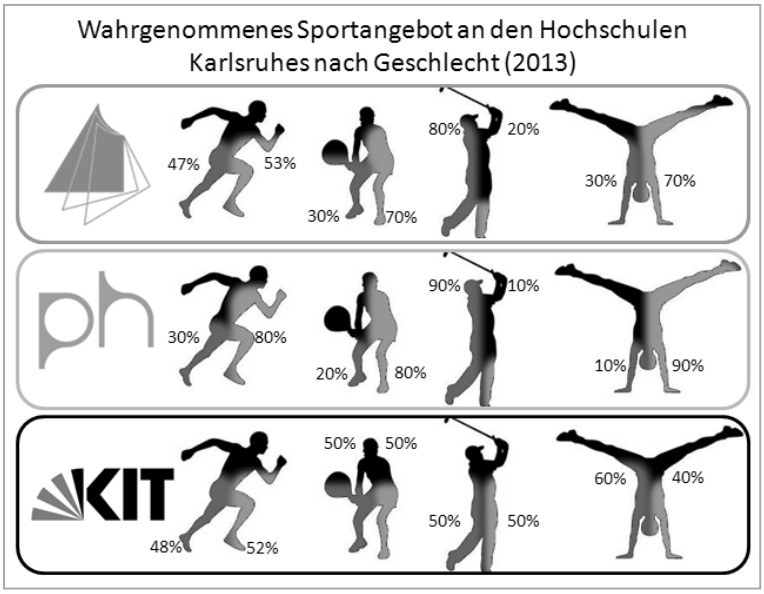


Bild 7.8 Eine Infografik mit bewusst reduzierter Beschriftung zeigt das Sportangebot der Hochschulen in Karlsruhe.

Bild 7.8 zeigt eine auf erdachten Daten erstellte Statistik über die wahrgenommenen Sportangebote an den verschiedenen Hochschulen in Karlsruhe. Dabei wird auf die namentliche/textuelle Nennung der Sportarten verzichtet. Diese wird stattdessen in Form von eingefärbten Silhouetten dargestellt. Ebenso werden die Namen der Hochschulen nicht ausgeschrieben, sondern durch die entsprechenden Logos ersetzt. Die Platzersparnis, die durch den Einsatz der Grafiken erreicht wird, hält sich hier sicher in Grenzen. Jedoch weiß der Betrachter sofort, dass die Daten im Diagramm thematisch dem Bereich Sport und Hochschulen zuzuordnen sind. Zu guter Letzt weckt eine grafische Repräsentation von Themen und Kontexten Aufmerksamkeit, und sie rufen weiterhin persönliche Assoziationen mit den eigenen Erfahrungen und Gefühlen des Betrachters hervor. Hat dieser etwa an der Hochschule Karlsruhe studiert, so entwickelt er direkt einen Bezug zu den präsentierten Werten. Die farbliche Kennzeichnung der Geschlechter könnte klassisch in den Farben Blau und Pink geschehen, sodass es keiner weiteren Erklärung der Farbgebung bedarf. Die prozentualen Angaben über die einzelnen Sportarten sind nicht unbedingt notwendig, um die Aussage der Grafik zu erschließen. So ist zum Beispiel bei bloßer Betrachtung der Silhouetten sofort zu sehen, dass Tennis weitestgehend von Frauen und Golf überwiegend von Männern gespielt wird. Wenn der Platz es jedoch zulässt und bestimmte Ausprägungen eines Merkmals teilweise schwer zu unterscheiden sind (z. B. Golf am KIT mit 50 % zu 50 %), dann kann eine Beschriftung einen vorteilhaften Effekt haben.

7.3.3 3D-Diagramme

Seit dem Erscheinen von *HTML5* und *WebGL* (*Web Graphics Library*) ist es in aktuellen Browsern möglich, aufwendige 3D-Grafiken zu rendern. So entstand beispielsweise die Bibliothek *PhiloGL* unter der Schirmherrschaft der Firma *Sencha*, die sich zwar nicht ausschließlich auf Datenvisualisierung beschränkt, aber dennoch einige interessante Ansätze zeigt, welche Art von Daten sich unter Zuhilfenahme einer dritten Dimension darstellen lassen. Wie so oft ist auch *Google* ganz vorne mit dabei, wenn es um moderne Web-Technologien geht, und stellt mit der für den hauseigenen Browser *Chrome* aufgesetzten Plattform *Chrome Experiments*[3] einige interessante Ansätze dar, die zeigen, welche Daten wie in 3D im Browser dargestellt werden können. Eines dieser Beispiele wurde als Inspiration verwendet, um später im praktischen Teil geologische Daten auf einer Weltkarte zu visualisieren.

Der Vorteil einer 3D-Darstellung liegt auf der Hand: Dem Benutzer wird nicht nur eine weitere Dimension geboten, in der er Daten ablegen und betrachten kann, vielmehr ist er außerdem in der Lage, viel intuitiver durch die drei Dimensionen zu navigieren. Natürliche Vorwärts- und Rückwärtsbewegungen ersetzen dabei beispielsweise einen Drilldown, um tiefer in Daten hineinzuschauen. Und, nicht zu vergessen, 3D wirkt immer innovativ und fortschrittlich! Gehen Sie mal über die *CeBIT*: Bei genauerem Hinsehen werden Sie von dreidimensionalen Darstellungen förmlich erschlagen.

Bild 7.9 3D-Sonographie eines Säuglings (© Valentina R. – Fotolia.com)

Eine praktische Anwendung findet die 3D-Darstellung bereits im medizinischen Sektor in den klassischen bildgebenden Verfahren wie etwa der *CT* (Computertomographie) oder der *MRT* (Magnetresonanztomographie). Dabei wird häufig aus einem Satz zweidimensionaler Bilder ein dreidimensionales Volumen rekonstruiert, das dann als 3D-Modell aus dem gewünschten Winkel betrachtet werden kann. Mitunter wird dem Verfahren der diagnostische Vorteil noch nicht zugesprochen, dient jedoch der Wissensvermittlung an Personen, denen die Daten nicht geläufig sind (Arzt ↔ Patient). So ist es z. B. für werdende Eltern viel interessanter, den Freunden und Verwandten eine Aufnahme ähnlich Bild 7.9 zu zeigen statt ein einfaches Ultraschallbild.

[3] http://www.chromeexperiments.com/

7.3.4 Ansätze, um Big-Data zu visualisieren

Aus den letzten drei Abschnitten ergeben sich nun einige Lösungsansätze, durch die wir auch große Datenmengen ansehnlich darstellen können. Diese sind neben den traditionellen Punkten, die Tidwell nennt, vor allem diese:

- *Interaktivität* – Erlauben Sie den Betrachtern, mit den Daten zu spielen, die Sichten zu verändern und verschiedene Kennzahlen und deren Beziehungen hervorzuheben. So gelingt es Ihnen nicht nur, mehrere Informationen zu kommunizieren, sondern Sie schaffen auch auf spielerischem Wege Motivation, das Diagramm weiter zu verändern und zu erforschen.
- *Dimensionen nutzen* – Es existiert ein Mythos, der besagt, dass ein Mensch lediglich 10 % der Leistung seines Gehirns benutzt. Auch wenn die Wissenschaft diese Aussage heutzutage widerlegt hat, hat sie doch in unserem Falle eine wahren Kern: Wir werden, zumindest beim Betrachten von Diagrammen, nicht besonders gefordert. Häufig offenbart sich uns der Sinn eines Balkendiagramms auf den ersten Blick, so betrachten wir etwa Umsätze eines bestimmten Zeitraums oder die Auslastung pro Mitarbeiter einer Abteilung und sehen sofort, welcher der umsatzstärkste Monat war oder welcher Mitarbeiter der *Mitarbeiter des Jahres* wird.

  Seien Sie bei der Auswahl Ihrer Diagramme mutig und verzichten Sie auf Balken- und Tortendiagramme. Wählen Sie stattdessen eines der im folgenden Abschnitt vorgestellten Charts und fordern Sie Ihre Mitarbeiter dadurch heraus, die Daten auch aus einer anderen, möglicherweise viel sprechenderen Sicht zu verstehen. Dieses *Fordern* kann dadurch geschehen, dass Sie neben der einfachen Zweidimensionalität und einer Farbgebung auf Basis von Grün, Gelb und Rot etwas mutiger werden und Bewegungen, zeitliche Veränderungen oder Dreidimensionalität in Ihre Visualisierung aufnehmen (auch hierzu folgen in Abschnitt 7.4 einige Beispiele). Geben Sie den Betrachtern mehr Zeit, um die Diagramme zu verstehen, und falls Sie die Diagramme nicht selbst erklären (etwa in einem Dashboard), ergänzen Sie ein Hilfesymbol, das die Verwendung des Diagramms erklärt. Sie werden sehen, dass Sie Ihr Publikum mit schicken, innovativen Diagrammen viel mehr begeistern als mit althergebrachten Grafiken. Verzichten Sie jedoch auf den Einsatz von Geräuschen, es sei denn, das Hauptaugenmerk Ihrer Datenrepräsentation liegt auf dem auditiven Kanal. Erstens haben Sie nicht immer die Möglichkeit, Geräusche überall abzuspielen, zweitens ist das Gehör der Menschen nicht immer gleich gut ausgebildet, um einzelne Nuancen der Daten zu erkennen, und drittens wirken Töne häufig störend bzw. lenken den Betrachter ab.

[4] http://datavisualization.ch/showcases/sound-mapping-in-new-york-city/

HINWEIS: Allerdings gibt es auch Beispiele, in denen eine Audiospur durchaus unterstützen kann. So stellt Mark Edward Campos beispielsweise das Leben in New Yorks Straße in einem 24-Stundenzyklus dar und zeigt die Öffnungszeiten verschiedener Einrichtungen an, etwa einer Bar, eines Cafés oder eines Fitnessstudios[4]. Er nutzt Bewegungen, um die Tageszeit in Stunden darzustellen, und eine Karte von New York im Hintergrund, um den ungefähren Standort des aktuellen Schauplatzes anzugeben. Campos setzt hier einen Audiokanal ein, um das Ambiente des jeweiligen Ortes widerzugeben. So hört man etwa im Park Vogelgezwitscher oder im Café sich unterhaltende Personen.

- *Kontextsensitivität* – Kontextsensitivität wurde bereits angesprochen, soll hier aber noch mal explizit als Möglichkeit genannt werden, Big Data verständlich zu visualisieren. Nutzen Sie in Ihrem Diagramm Grafiken, die darauf hindeuten, in welchen Themenbereichen sich die Daten bewegen. So können Sie etwa Logos von Automarken in einem Diagramm als Marker verwenden und Beziehungen durch Straßen darstellen. Somit wecken Sie Interesse und müssen Datensätze nicht explizit beschriften, wodurch Sie wiederum Platz sparen.

Behalten Sie diese drei Punkte einmal im Hinterkopf, wenn wir in den folgenden Abschnitten ein Visualisierungsframework auswählen, und überprüfen Sie diese Diagrammtypen auf das Vorhandensein von interaktiven Elementen, kontextsensitiven Darstellungen und die Verwendung verschiedener Dimensionen in Form von Bewegung, Zeit und Schichten. Sie werden sehen, dass diese häufig Verwendung finden.

TIPP: Die Ansätze, zu denen ich hier rate, sind natürlich nicht universell gültig und in gewisser Hinsicht auch von Erfahrungen und Geschmäckern einzelner Personen oder Unternehmen abhängig. Nicht jedem gefallen bunte, sich bewegende Diagramme, manche wünschen sich etwas mehr Konservativität und Gewöhnlichkeit. Hier eine kleine Anekdote von einer Präsentation, die ich auf einem Workshop 2014 zum Thema *Big-Data-Visualisierung* gehalten habe.
Mein Vorredner referierte sehr kompetent und kurzweilig über *Visual Business Analytics*, riet jedoch bei animierten, *glitzernden* und interaktiven Diagrammen zur Vorsicht. Ich zeigte hingegen direkt im Anschluss vor allem bunte, ausgefallene, komplexe und interaktive Diagramme. In der ersten Pause kamen die Zuschauer auf die Redner zu und unterhielten sich über das Gesehene. Ich kam schnell mit ein paar Studenten und jungen Entwicklern ins Gespräch, mein Vorredner hingegen mit Leuten aus der Forschung, von Banken und Professoren. Damit will ich sagen, dass es kein *Richtig* oder *Falsch* gibt, sondern dass es immer ganz darauf ankommt, welches Publikum man mit seinen Diagrammen anzusprechen gedenkt.

11 Polyglot Persistence

Relationale Datenbanksysteme galten bis in die Milleniumsjahre als unangefochtener Standard, um Daten abzulegen und auszuwerten. Wurde früher in Schulen und Universitäten das Fach Datenbanken unterrichtet, ging es fast ausschließlich um Themen wie Normalisierung, relationale Algebra und das ACID-Modell. Standards abseits von relationalen Datenbanksystemen wie objektorientierte Datenbanksysteme galten als Nischenprodukte.

Martin Fowler propagierte 2011 den Gedanken der Polyglot Persistence (Fowler, 2011). Dieser Begriff besagt, dass für spezifischere Anforderungen auch spezifischere Datenbanksysteme benötigt werden. Er folgerte daraus, dass es eine große Bandbreite an Technologien für unterschiedliche Probleme geben wird.

Über die drei Vs, die in den ersten Kapiteln thematisiert wurden, können Anforderungen skizziert werden, die mit relationalen Systemen nicht oder nur teilweise gelöst werden können. Polystrukturierte Daten passen im Regelfall nicht in zweidimensionale Schemata. Ausnahmen sind relationale Datenbanksysteme, die spezielle Spaltentypen anbieten. Im Regelfall erfordert es eine Kompromisslösung, wenn JSON- oder XML-Dokumente in einer Spalte abgelegt werden sollen. Eine dieser Kompromisslösungen wäre, die Daten als BLOB-Objekt zu speichern und erst beim Auslesen zu interpretieren. Das kann für manche Anwendungsfälle zu langsam sein. Zudem braucht man auch eine API, die das Verarbeiten und Interpretieren der Daten übernimmt.

Eine andere Lösung wäre es, polystrukturierte oder speziell codierte Daten mittels ETL-Werkzeuge aufzubereiten und in eine relationale Datenbank zu laden. Doch hier gibt es zwei Probleme:

- *Performanz:* Bei komplexen Formaten und großen Datenmengen kann das ETL-Werkzeug zum Flaschenhals werden. In einem kontinuierlichen Datenstrom entsteht das Risiko, dass mehr Daten angeliefert als verarbeitet werden können.
- *Duplizierung:* Es kann Applikationen in der IT-Landschaft geben, die die Daten im Ursprungsformat verarbeiten. So müssten die Daten wieder deserialisiert werden, was Zeit kostet und aufwendig sein kann. Eine Alternative wäre, die Daten zweifach zu speichern, was wiederum zu Synchronisationsproblemen führen kann.

Immer mehr Datenbanksysteme und Datenplattformen, die oft auf verteilten Dateisystemen aufbauen, kommen auf den Markt. Antreiber für diese Entwicklung sind neue Features, die für Applikationen benötigt werden, aber auch neue Infrastrukturmöglichkeiten wie etwa die Cloud. Polyglot Persistence ist ein Trend, der seit der ersten Auflage des Buches an

Bedeutung gewinnt: Hierbei stellt sich weniger die Frage, ob Hadoop oder ein relationales Datenbanksystem verwendet werden soll. Auch ob man sich auf unstrukturierte oder strukturierte Daten konzentriert, ist vielleicht nicht mehr die zentralste Frage. Die Kernfrage ist, welches System am besten auf die Anforderungsmatrix passt, die von Anwendungsfällen abgeleitet werden können.

Folgende Trends sind bei den Datenplattformen zu sehen:

- **Dateisysteme:** HDFS ist vielleicht das bekannteste verteilte Dateisystem. Das liegt an der hohen Verbreitung von Apache Hadoop. Es gibt jedoch in der Kategorie DFS (Distributed File System) auch zahlreiche Alternativen. Vertreter davon sind Systeme wie z. B. HopFS (The Morning Paper, 2017). Im Hadoop-Ecosystem ist Mapr-XD (vormals MapR-FS) eine interessante Alternative zu HDFS, da sich MapR-FS durch seine Architektur auch besser für kleine Dateien eignet (Drushal, 2017).
- **Relationale Datenbanken:** Auch wenn die Marktführer am Markt wie Oracle und Teradata(Teradata, 2018) ihre Systeme mittlerweile auch in der Cloud anbieten, gibt es zahlreiche Anbieter, die ihre Systeme als native Cloudlösung anbieten und somit neue Dynamiken in den Markt bringen. CockroachDB und Snowflake sind beides Datenbanken, die immer wieder in Verbindung mit Cloud-Infrastrukturlösungen genannt werden. Natürlich darf hier auch Redshift von Amazon nicht unerwähnt bleiben.
- **NoSQL:** Die in Kapitel 5 bereits vorgestellten NoSQL-Systeme werden immer wieder durch neue Systeme erweitert. Auch hier gilt, dass Cloud-Lösungen neuen Schwung bringen können, da Marktführer wie MongoDB, Redis und andere Systeme nicht Cloud-nativ sind.
- **Cloud-spezifische Massenspeicher:** Cloud-native Lösungen wie Amazon S3 werden bei Firmen immer populärer, um Massendaten zu speichern. Die Idee ist, einen Objektstore zu verwenden, bei dem das Speichern von Daten mit geringen Kosten und hoher Verfügbarkeit verbunden ist. Datenverarbeitungsengines wie Spark oder SQL-Engines wie Impala sind auch in der Lage, Daten direkt von S3 zu laden.

11.1 Praxis

Im Folgenden werden verschiedene Systeme anhand von Beispielen vorgestellt, damit sie diese besser evaluieren können.

11.1.1 Redis

Key Value Stores wurden in Kapitel 5 vorgestellt. Ein Vertreter dieser Kategorie ist Redis. Der Hersteller von Redis gibt fünf Hauptanwendungsgebiete an (Redis, 2017):

- User Session Data Management
- Echtzeitanalysen wie Zähler und Ranglisten

- Empfehlungen wie Kauf- oder Artikelempfehlungen aufgrund von Profilcharakteristika.
- Message Queues für Workflows und andere Jobs
- Caching von statischen und interaktiven Daten

  Sie werden sich nun vielleicht fragen, was die Verbindung zu Big Data ist. Redis eignet sich perfekt dazu, Ergebnisse aus Big-Data-Analysen zwischenzuspeichern. Key Value Stores wie Redis sind somit für den Einsatz in großen Big-Data-Landschaften gut geeignet.

Die folgenden Zeilen erstellen einen Docker-Container mit Redis. Als Erstes wird ein Docker-Container erstellt und gestartet. Das Docker-Image stammt vom Hersteller selbst.

```
docker run --name redis -d redis
docker exec -it redis bash
```

Danach sollten Sie mit einem Docker-Container verbunden sein, in dem Redis läuft.

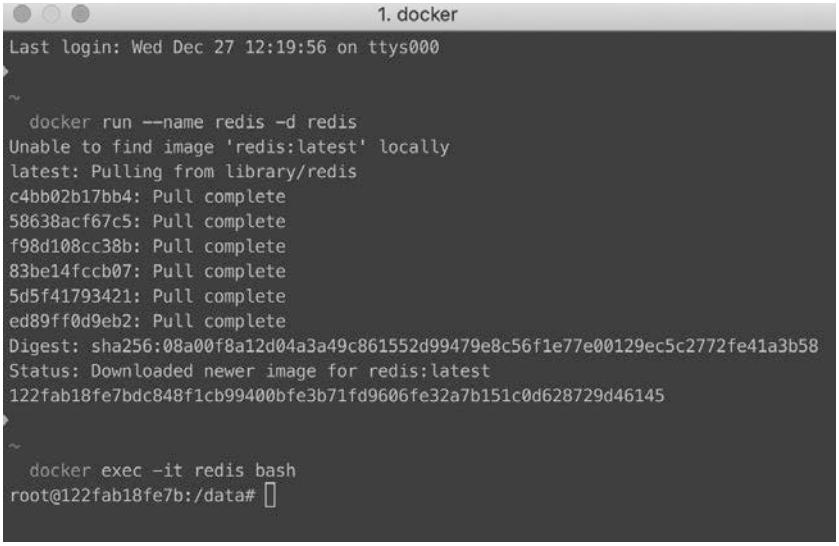


Bild 11.1 Der Docker-Container von Redis wird gestartet.

Die folgenden drei Befehle starten einen Server. Der zweite Befehl ist ein Ping, und mit dem letzten Befehl wird der Client gestartet. Diese Befehle führen Sie bitte in der aktiven Konsole aus.

```
redis-server
redis-cli ping
redis-cli
```

Nach dem Ausführen dieser Befehle sollte die Ausgabe dem folgenden Screenshot ähneln.

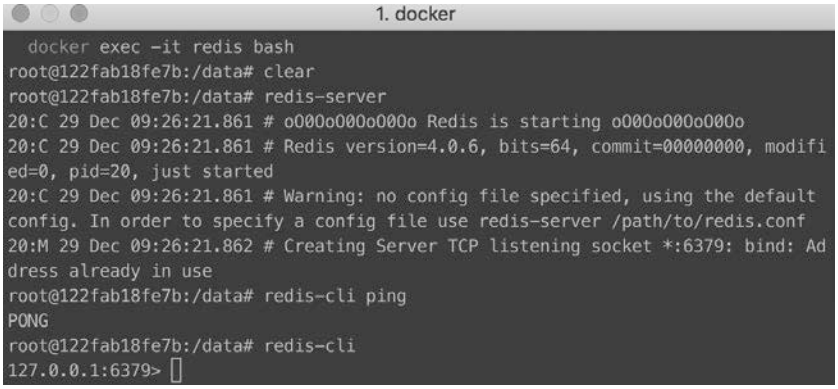


Bild 11.2 Die Redis-Kommandozeile

Nachdem der Client gestartet ist, setzen Sie einen einfachen Wert:

```
set mykey somevalue
get mykey
```

Mit diesen beiden Befehlen decken Sie bereits den Kernanwendungsfall eines Key Value Store ab: Sie setzen einen Wert und rufen ihn wieder ab.

Ein beliebter Anwendungsfall von Key Value Stores sind Zähler und Ranglisten.

```
set counter 100
incr counter
```

Wer Redis gerne auch einmal mit Python ansprechen will, kann folgende Befehle ausführen:

```
apt-get update
apt-get install -y python python-pip
sudo pip install redis
```

Wer dann den Python-Interpreter startet, kann die Python-REPL nutzen, um folgende Befehle auszuführen:

```
>>> import redis
>>> r = redis.StrictRedis(host='localhost', port=6379, db=0)
>>> r.set('foo', 'bar')
>>> r.get('foo')
```

Diese kleine Einführung sollte ausreichen, um weiter mit Redis zu arbeiten. Im Internet finden Sie zahlreiche Anleitungen, wie Sie mit der Programmiersprache Ihrer Wahl Daten in Redis verarbeiten können.

11.1.2 MongoDB

MongoDB ist eine freie, plattformübergreifende Dokumentendatenbank (MongoDB, 2018)

Folgende Zeilen sind der schnellste Weg zum Testsystem:

```
docker run --name mongo -d mongo
docker exec -it mongo bash
```

Der Mongo-DB-Client wird gestartet, und einzelne Befehle werden ausgeführt:

```
mongo
db.literature.insert([
   {
      title: 'Hadoop - Definitive Guide',
      author: 'Tom White',
      publisher: 'OReilly Media',
      url: 'http://shop.oreilly.com/product/0636920033448.do',
      tags: ['hadoop', 'hive', 'yarn'],
      releasedate: 'April 2015',
      Pages: 235
   },
   {
      title: 'PolyglotPersistence',
      description: "An article on polyglot persistence",
      author: 'Martin Fowler',
      url: 'https://martinfowler.com/bliki/PolyglotPersistence.html',
      tags: ['rdbms', 'mongodb', 'neo4j'],
      releasedate: '16. November 2011'
   }
])
db.literature.find().pretty()
```

Obiges Beispiel veranschaulicht exzellent den klassischen Anwendungsfall eines Literaturverzeichnisses. Literaturverzeichnisse können unterschiedliche Felder haben und sind somit als Schema schwer fassbar. Ist dieses Schema in JSON in MongoDB festgehalten, ist es auch leicht möglich, die Daten zu extrahieren und als JSON-Dokument an Applikationen zu übergeben.

11.1.3 Neo4j

Neo4j ist eine Graphendatenbank, die sich für die Darstellung von Beziehungen eignet. Sie gilt als Standard für alle Formen von Graphen, wie z. B. die Darstellung von Wegverbindungen. Neo4j kann über folgenden Befehl gestartet werden:

```
docker run \
    --publish=7474:7474 --publish=7687:7687 \
    --volume=$HOME/neo4j/data:/data \
    neo4j
```

Nun kann im Browser auf *http://localhost:7474* gewechselt werden, wo man sich mit dem User *neo4j* und dem Passwort *neo4j* in der WebUI einloggen kann. Dort findet man auch ein einfaches Tutorial zu neo4j.

11.1.4 S3

Sie kennen wahrscheinlich die berühmte Gretchen-Frage aus Goethes *Faust*. Abgeleitet davon kann man auch vielen IT-Leitern die berühmte Frage stellen: „Nun sag, wie hast du's mit der Cloud?"

Die Antworten könnten nicht unterschiedlicher sein. Viel hängt auch davon ab, in welchen Branchen die Unternehmen tätig sind. Auch Kultur spielt eine Rolle. Im angloamerikanischen Raum scheint man Cloud-Technologien gegenüber offener zu sein als im zentraleuropäischen Bereich.

Einige Unternehmen halten am Grundsatz fest: „Wir geben doch unsere Daten keinem amerikanischen Unternehmen, wo wir gar nicht wissen, was da passiert!" Andere beginnen, sich nach und nach zu öffnen. Lassen Sie uns deswegen einen Blick auf Amazon S3 werfen. Neben Amazon S3 gibt es natürlich auch andere Cloud-Anbieter, die eine Art Massenspeicher in der Cloud anbieten, der wie ein Filesystem genutzt werden kann.

Die Daten, die auf S3 gespeichert werden, haben eine garantierte Verfügbarkeit von 99,999999999 %. Auf *https://aws.amazon.com/s3/pricing/* können die Preise abgerufen werden. Mit Stand beim Schreiben dieses Buchs kostete ein Gigabyte 0,0245 US$. Ein Terabyte für 24,50 Euro ist für eine Firma kein großes Problem. Würden eine Firma ein Terabyte an Daten selbst hosten, müsste sie Infrastruktur einkaufen und IT-Mitarbeiter einstellen, die den Betrieb gewährleisten. Selbst bei billigster Hardware und geringsten Gehaltskosten ist offensichtlich, dass internen IT-Abteilungen ein ähnliches Schicksal drohen könnte wie dem Tante-Emma-Laden um die Ecke.

Für viele Kunden stellt sich also durchaus die Frage, warum nicht Daten auf Massenspeicher wie Amazon S3 hochladen und das Filesystem wie HDFS nutzen. SQL-Engines wie Impala können auch S3-Daten abfragen, und auch Apache Spark verfügt über Schnittstellen zu S3.

Berücksichtigen sollte man bei der Entscheidung für S3 allerdings Sicherheits- und Performanceaspekte. Ein Kompromiss, den viele Unternehmen dann in Erwägung ziehen, ist ein Hybridbetrieb. Amazon S3 wird manchmal als sogenanntes Cold Storage genutzt, um Daten zu archivieren. Auch die Durchführung von sogenannten POCs (Proof of Concepts) in der Cloud kann für viele Unternehmen reizvoll sein.

Wer Daten auf Amazon Web Services (AWS) ablegen möchte, muss sich einen Account anlegen. Dabei kommt wieder das leidige Thema auf, dass man bei Cloud-Anbietern seine Kreditkartendaten hinterlegen muss. Ich verstehe vollkommen, dass das nicht jeder will. Falls dieser Schritt – Ihre Finanzdaten einzugeben – für Sie in Frage kommt, können Sie, nachdem Sie Ihren Account angelegt haben, ein paar der folgenden Schritte selbstständig nachspielen. Die folgenden Schritte sollten selbsterklärend sein.

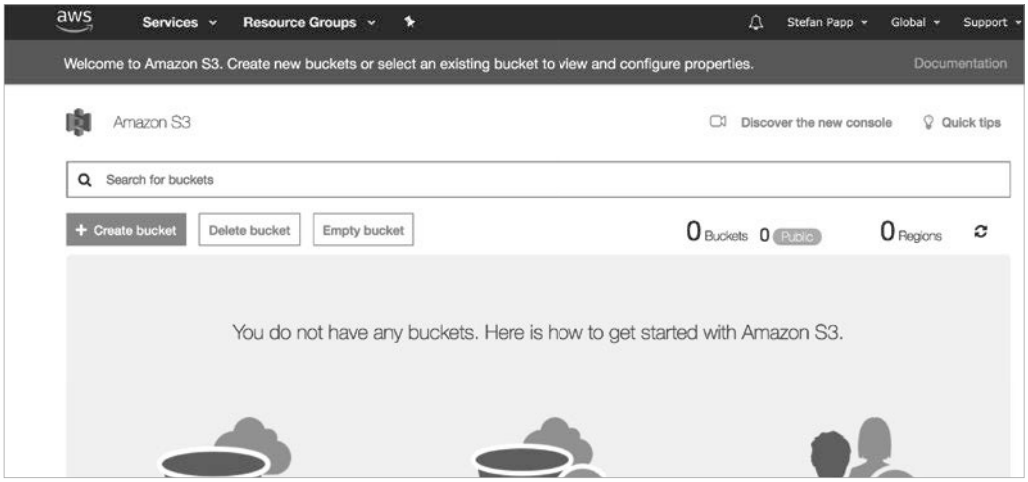


Bild 11.3 AWS-Benutzeroberfläche

Bild 11.3 zeigt die Amazon S3-Oberfläche. Sie können sogenannte *Buckets* anlegen (siehe Bild 11.4), also Container, in denen man ähnlich wie in Verzeichnissen Daten ablegen kann.

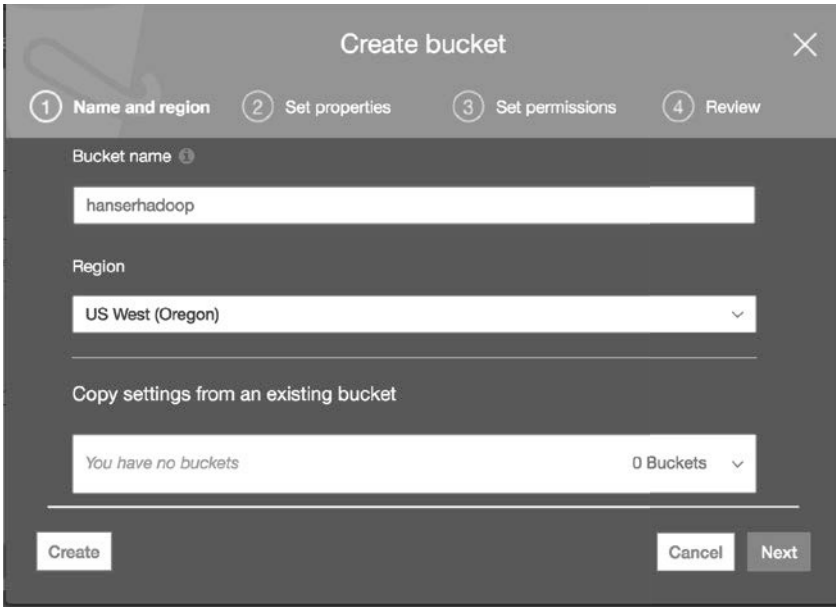


Bild 11.4 Erstellen eines Buckets

Im Schnelldurchlauf kann man einige Parameter durchgehen, die ebenfalls selbsterklärend sind, um die Schritte in der AWS-UI abzuschließen. Man kann festlegen, wer auf die Buckets Zugriff bekommt, und auch eine Versionierung ist konfigurierbar. Wesentlich ist, einen eindeutigen Namen festzulegen und bei der Namensfindung kreativ zu sein, da bei vielen weltweiten Nutzern natürlich auch viele eindeutige Namen bereits vergeben sein können.

Um dann vom lokalen Dateisystem auf die Cloud zugreifen zu können, müssen Sie auch noch unter *https://console.aws.amazon.com/iam/home?#/security_credential* Ihre Security Credentials konfigurieren.

Wer von der Kommandozeile Daten in die Cloud hochladen will, muss eine API installieren, die man über die Konsole auch aufrufen kann. Die AWS CLI ist in Python geschrieben und kann auch über den Python-Installer installiert werden:

```
pip install awscli
```

Danach gilt es, die CLI zu konfigurieren. Wesentlich dabei ist, die davor generierten Schlüssel zu setzen. Dies geschieht über

```
aws configure
```

Ich habe im Vorfeld, wie man im Screenshot sehen kann, einen Bucket namens *hanserhadoop* angelegt. Diesen kann ich nun abfragen:

```
aws s3 ls s3://hanserhadoop
```

Um zu zeigen, dass alles funktioniert, legen wir eine leere Datei an und laden sie auf AWS hoch. Danach führen wir den *ls*-Befehl erneut aus:

```
touch test.txt
aws s3 cp test.txt s3://hanserhadoop
aws s3 ls s3://hanserhadoop
```

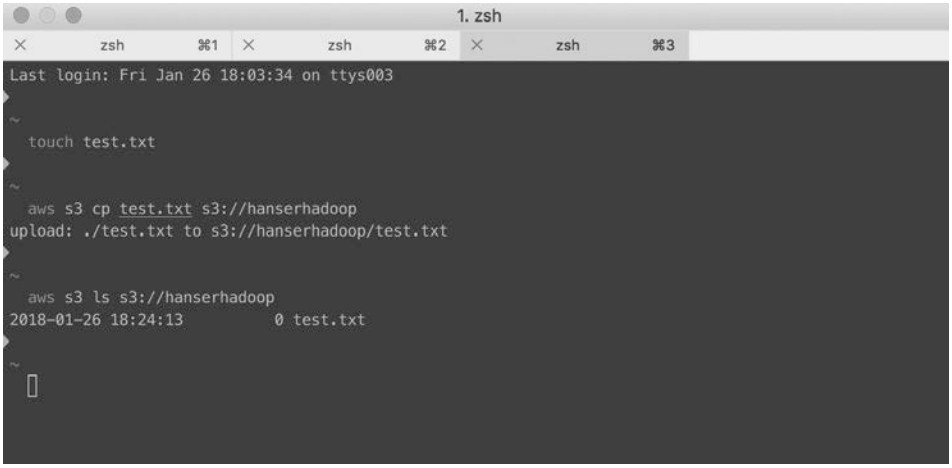


Bild 11.5 AWS-Befehle auf der Kommandozeile ausführen

Natürlich sieht man die hochgeladene Datei dann auch in der Web-Oberfläche.

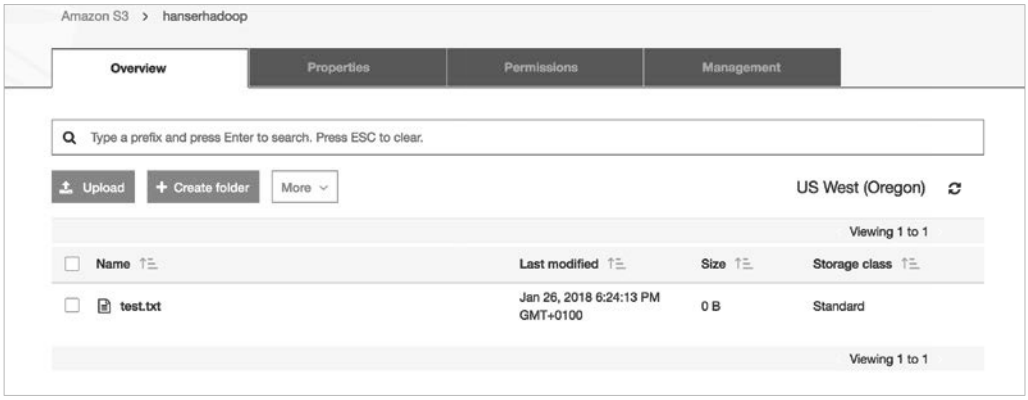


Bild 11.6 AWS Web-Oberfläche

Die Daten in der Cloud lassen sich also von den eigenen Systemen aus abfragen. Die logische Konsequenz ist, dass wir alle Aktionen, die wir auf einem verteilten Dateisystem wie HDFS ausgeführt haben, auch auf S3 ausführen können.

11.1.5 Apache Kudu

In diesem Buch wurde als spaltenorientierte Datenbank HBase bereits hinreichend dargestellt. Es macht jedoch Sinn, eine Alternative zumindest zu erwähnen.

Apache Kudu ist ein Projekt, das ein Hybrid zwischen einer Datenbank und dem Filesystem sein will.

Leider gab es zum Zeitpunkt des Verfassens des Buchs kein funktionierendes Docker-Image. Deswegen kann ein Quickstart nur unter *https://kudu.apache.org/docs/quickstart.html* abgerufen werden.

11.2 Zusammenfassung

In diesem Kapitel wurde gezeigt, dass es für jeden Anwendungsfall ein eigenes System gibt, um Daten abzulegen und zu analysieren. Die Ecosysteme ändern sich rasant und Sie haben einen Vertreter aus jeder Kategorie kennegelernt.

Wichtig ist, die Technologie immer den spezifischen Anforderungen entsprechend auszuwählen. „Glaubenskriege“ darüber welche Technologie die beste sei, sind hier meist nicht besonders hilfreich.

Index

Symbole

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z