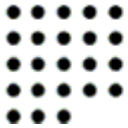

Fachhochschule Köln
Cologne University of Applied Sciences

Bachelorarbeit (Wirtschaftsinformatik BA)

Big Data und ihre Technologien:

Hadoop und NoSQL-DBMS

Erstprüferin: **Prof.Dr.Birgit Bertelsmeier**

Zweitprüferin: **Prof.Dr.Heide Faeskorn-Woyke**

Ausgearbeitet von:

Serafettin Coskun

e-mail: serefcoskun@msn.com

vorgelegt an der
Fachhochschule Köln, Campus Gummersbach
Fakultät für Informatik und Ingenieurwissenschaften

Gummersbach, Januar 2014

Abstract

The following Bachelor Thesis deals with Big Data and its related technologies, which are NoSQL and Hadoop. The selection of the topic was made deliberately, because of its actuality as well as its growing importance from a business perspective. Due to ever-growing data, that is available up to 80% in semi-structured form, the IT infrastructure of a company quickly reaches its limits. Exactly at this point Big Data technologies like NoSQL and Hadoop should help to process large amounts of data in order to provide valuable information that was not previously recognized.

In dieser Bachelorarbeit wird das Thema Big Data und die damit verbundenen Technologien, sprich NoSQL und Hadoop behandelt. Das Thema wurde bewusst ausgewählt, weil sie zum einen aktuell und zum anderen immer mehr an Bedeutung, aus Sicht der Unternehmen gewinnt. Durch die ständig anwachsenden Daten, die zu 80% in Semistrukturierter Form vorliegen, stößt die IT-Infrastruktur eines Unternehmens schnell an seine Grenzen. Und genau an dieser Stelle sollen die Big Data Technologien, wie „NoSQL“ und „Hadoop“, helfen, die so großen Datenmengen zu verarbeiten, um für das Unternehmen, wertvolle Informationen zu liefern, die bislang nicht ersichtlich waren.

Inhaltsverzeichnis

1.Einführung

Da heute in einer Digitalen Welt gelebt wird, werden ungeheure Mengen an Daten erzeugt. Bislang haben sich die Unternehmen mit Transaktionsdaten, also den strukturierten Daten befasst. Diese haben sie sich zu Nutze gemacht, und bei Entscheidungen für das Unternehmen genutzt. Strukturierte Daten sind Daten, die man gezielt nach einzelnen oder zusammengesetzten Attributen suchen bzw. sortieren kann.[1]

Durch die immer grösser werdende Soziale Vernetzung, wie z.B. durch Soziale Medien, wie Twitter, Facebook, e-mails, Web blogs etc., entstehen zusätzlich Terabyte, Hexabyte an Daten. Auch mobile Geräte, Kameras und Sensoren sind an diesen Daten beteiligt. Die Eigenschaft dieser Daten besteht darin, dass sie in einer unstrukturierten Form vorliegen. D.h., dass die Daten nicht in einer vordefinierten, strukturierten Tabelle gespeichert sind. Sie bestehen meist aus Zahlen, Texten und Faktenblöcken, und weisen kein spezielles Format auf. Zu 80% der Daten, die heute erzeugt werden, sind unstrukturiert. Allein die Menge dieser Daten ist sehr gigantisch. Z.B. die Musikdateien, die es bei iTunes gibt. Es sind über 28millionen Songs und über 30billionen Downloads. Wenn man nur von Social Media ausgehen würde, gibt es allein bei Twitter jeden Tag über 500Millionen Tweets. Die Vielfalt dieser Daten spricht auch für sich, Text, Protokolldaten, Video, Audio, Klickstreams etc., sind ihre Bestandteile (Siehe Kapitel 2.1). [2]

Laut einer Studie von Intel werden die Daten bis 2015 um 5 Hexabyte ansteigen, und bis zum Jahre 2020 um das 40 fache.[3]

Die Idee von Big Data steckt darin, aus so einer großen Datenmenge, sinnvolle Gewinnung und Nutzung entscheidungsrelevanter Erkenntnisse, aus qualitativ vielfältigen und unterschiedlich strukturierten Informationen, die einem schnellen Wandel unterliegen, und in bisher ungekanntem Umfang anfallen, zu erlangen, um für das Unternehmen Wettbewerbsvorteile und Wirtschaftliche Nutzen zu verschaffen. Die Idee von Big Data besteht aber auch darin, durch neue Analysemethoden, neue

[1] vgl. [GK_2013] Kap.1, S.240
[2] vgl. [ZK_2013] Kap.1, S.1
[3] vgl. [INT_2013]

Informationen zu gewinnen, um bisher unbekannte Zusammenhänge, auf Basis der Unmengen an Daten, die zugrunde liegen, sichtbar zu machen.[4]

Die für diese Bachelorarbeit anfallenden Themen, ergeben sich aus der Realisierung von Big Data im Unternehmen. Natürlich können nicht alle Aspekte bearbeitet werden, da der Umfang, und die Zeit für die Bachelorarbeit auch begrenzt ist. Die angesprochenen Themen sind folgende:

- Was ist Big Data?/Definition(siehe Kapitel 2.1)
- Vorteile und Potentiale/Probleme, Herausforderungen(siehe Kapitel 2.2-2.4)
- Datenschutz(siehe Kapitel 2.5)
- Big Data Analytik (Kapitel 3)
- Big Data Technologien, Hadoop, NoSQL-DBS etc. (Kapitel 4 u.5)
- Fazit

[4] vgl. [BIT_2012] Kap.1, S.7

2 Erläuterungen

In diesem Abschnitt wird Big Data definiert (siehe Kapitel 2.1), die Vorteile sowie die Potentiale (siehe Kapitel 2.2) von ihr erläutert. Die Einsatzgebiete von Big Data werden aufgegriffen und stichpunktartig aufgelistet und erklärt(siehe Kapitel 2.3). Da Big Data nicht nur Vorteile mit sich bringt, werden die Probleme und auch die Herausforderungen erläutert(siehe Kapitel 2.4). Dabei wird der Aspekt Datenschutz hervorgehoben, und mit einem Unterkapitel versehen(siehe Kapitel 2.5).

2.1 Big Data Definition

Zunächst einmal sollte betont werden, dass es keine einheitlich Definition für Big Data gibt. Jeder definiert Big Data nach seiner eigenen Priorität, und für das Unternehmen relevanten Aspekten. Da hier nicht alle Aspekte angesprochen werden können, wird an dieser Stelle eine Kombination von Definitionen zusammengestellt, die weitestgehend mit den Themen dieser Bachelorarbeit im Einklang stehen.

Big Data bezeichnet das schnelle Anwachsen von Daten, die sowohl in strukturierter Form von Zahlen und Tabellen, als auch in unstrukturierter Form von Texten, Webseiten und Videos vorliegen. Einer der großen Merkmale dieser Daten ist es, dass sie sich mit hoher Geschwindigkeit verändern.[5]

Big Data bezeichnet aber auch den Einsatz von großen Datenmengen, die aus verschiedenen Quellen gewonnen werden, um wirtschaftlichen Nutzen zu erlangen. Sie umfasst also jede Art von Daten, die in Beziehung zum Geschäftsmodell des Unternehmens stehen, und die Möglichkeit, sie zu analysieren. Daraus ergibt sich, dass Big Data jegliche Daten umfasst, und strukturierte von unstrukturierten Daten nicht unterscheidet.[6]

Wenn von unstrukturierten Daten die Rede ist, kann diese Situation auch von einer anderen Perspektive betrachten werden. Es gibt einige, die denken, dass Unternehmen, Daten deswegen als unstrukturiert definieren, weil die neuen Daten nicht konform mit dem vordefinierten Format, Modell oder Schema ist. Z.B. könnte der body einer e-mail als unstrukturiert betrachtet werden, sie ist aber ein Teil einer bestimmten Struktur, das den Spezifizierungen von RFC-2822(Internet Message Format) entspricht, und

[5] vgl.[SAS_2013]
[6] vgl.[GK_2013] Kap.1, S.240

eine Reihe von Feldern enthält, die Von, Bis, Thema und Datum umfassen. Im Jahre 2001 hat Doug Laney von der Meta Group (eine IT-Forschungsgesellschaft, die von Gartner 2005 erworben ist), eine Forschungsarbeit geschrieben, in der er festgelegt, dass e-commerce die Datenverwaltung entlang drei Dimensionen gesprengt hat:[7]

- **Volume**(Datenmenge). Da es sich bei Big Data um große Datenmengen handelt, haben die Unternehmen meist ein Problem, wenn man alle Daten im Unternehmen nutzen will.
- **Velocity**(Geschwindigkeit bzw. Verfügbarkeit). Die Daten aus Big Data müssen meist oft sofort, oder zumindest zeitnah verarbeitet werden, wenn sie in das Unternehmen einfließen, damit sie optimal für das Unternehmen genutzt werden können.
- **Variety**(Vielfalt). Big Data beinhaltet nicht nur strukturierte Daten, sondern umfasst darüber hinaus noch unstrukturierte Daten: Text, Protokolldaten, Video, Audio, Klickstreams… Mehr als 80 Prozent der Daten sind heute unstrukturiert.[8,9]

Durch den Punkt Variety, sieht man aber auch, dass die Quellen von Big Data sehr vielseitig ist:

- Unternehmensdaten(emails, word Dokumente)
- Transaktionsdaten
- Social Media
- Sensordaten
- Öffentliche Daten(Energie, Welt Ressourcen, Laborstatistiken)
- Wissenschaftlich Daten, die sich auf das Wetter und auf die Atmosphäre beziehen
- Daten, die durch medizinische, ärztliche Verfahren, wie Radiologie oder CT scan etc. entstehen.
- Bilder und Videos
- RFD(Radio Frequency Data)

[7] vgl.[ZK_2013] Kap.1, S.1
[8] vgl.[GK_2013] Kap.1, S.241
[9] vgl.[ZK_2013] Kap.1, S.1

Abgesehen von Schwierigkeiten, solche Daten zu managen und zu speichern, ist es auch eine Herausforderung, diese abzufragen, zu analysieren und zu visualisieren.[10]

Diese Kriterien werden die drei Vs, oder als V3-Kriterien von Big Data genannt. Es gibt aber auch Quellen, wo es das vierte V-Kriterium gibt:

- **Veracity**(Wahrhaftigkeit), bedeutet im allgemeinem Wahrhaftigkeit oder die Einhaltung, das Befolgen der Wahrhaftigkeit. In Bezug auf Big Data, ist es mehr eine Herausforderung, als eine Eigenschaft. Es ist schwierig, aus einer riesigen Datenmenge die Wahrhaftigkeit festzustellen, die eine hohe Velocity(Geschwindigkeit) hat. Es besteht immer die Möglichkeit, ungenaue, und unsichere Daten zu haben. Daher ist es eine herausfordernde Aufgabe, solche Daten zu reinigen, bevor sie erst analysiert werden.[11,12]

Volume ist aber der erste Gedanke, der mit Big Data aufkommt. Manche betrachten Petabytes, als den Startpunkt von Big Data. In bestimmten Fällen kann es aber auch sein, dass der Nutzen von Daten auch in einer begrenzten Datenmenge liegen kann, wenn diese einen hohen Wert für das Unternehmen haben. Zu male, seit langer Zeit riesige Mengen an Transaktionsdaten verarbeitet werden, ohne, dass sie als Big Data bezeichnet wurden. Solange aber nicht mehrere dieser V-Kriterien erfüllt sind, ist von Big Data erst gar nicht die Rede.

Eng betrachtet impliziert der Begriff Big Data große Datenmengen. Aber nur mit Daten kommt man nicht weiter. Denn was bringen so viele Daten, wenn es keine passenden Analysemethoden gibt. Bei Big Data handelt es sich nicht um:

- eine einzelne Technologie
- ein Hardware- oder Softwareprodukt
- eine IT-Strategie oder Architektur
- ein Marketing-Buzzword
- oder einen kurzfristigen Trend.

[10] vgl.[WM_2013]
[11] vgl.[WM_2013]
[12] vgl.[BIT_2012] Kap.4, S.19

Bei Big Data handelt es sich um eine Erweiterung, von schon vorhandenen Datenanalysemethoden, -prozessen und –verfahren (siehe Kapitel 3).[13]

2.2 Vorteile und Potentiale von Big Data

Laut einer Studie von BARC(Business Application Research Center), erwarten die Unternehmen, die meisten Vorteile von Big Data, auf strategischer Ebene.[14] Doch um wirklich den Best möglichen Nutzen zu erlangen, ist es sehr wichtig, dass Fachleute aus unterschiedlichen Bereichen, Hand in Hand arbeiten. Dazu gehören IT-Fachleute, Business Manager und Experten für das Sammeln und Auswerten von großen Datenbeständen. Zu den wichtigsten Vorteilen von Big Data zählen folgende Aspekte:

- Bessere strategische Entscheidungen
- Bessere Steuerung operativer Prozesse
- Schnellere und detailliertere Analysen von Daten
- Verbesserter Kundenservice
- Zielgerichtetere Marketingaktionen
- Besseres Verständnis des Marktes/Wettbewerbs
- Geringere Kosten
- Bessere Produkt-/Service-Qualität
- Bessere Kundenbindung.

Die professionelle Analyse von Big Data kann Wettbewerbsvorteile, Einsparpotenziale und sogar neue Geschäftsfelder einschließen. Doch um diese Vorteile zu erlangen, reichen technische Fertigkeiten und fachliches Wissen nicht aus. Big Data erfordert fachbereichsübergreifendes denken, das Informationen vormals klar abgegrenzten Bereichen zusammenführt.[15]

Wenn Big Data destilliert und in Kombination mit den traditionellen Unternehmensdaten analysiert sind, können Unternehmen ein gründlicheres, und aufschlussreicheres Verstehen ihres Geschäfts entwickeln. Dies kann zu einer erhöhten Produktivität, einer stärkeren Wettbewerbsposition, und größeren Innovation führen, von denen alle einen bedeutenden Einfluss auf das Endergebnis haben können. [16]

[13] vgl.[GK_2013] Kap.1, S.241
[14] vgl.[IBMG_2013]
[15] vgl.[IBMG_2013]

2.3 Einsatzgebiete von Big Data

Big Data könnte in vielen Bereichen eingesetzt werden. Hierzu werden ein paar Beispiele angeführt:

- Das Gesundheitswesen wäre ein Beispiel dafür. Zum Beispiel, in den Liefergesundheitsfürsorge-Dienstleistungen, ist das Management chronischer, oder langfristiger Bedingungen teuer. Der Gebrauch von Mithörgeräten im Haus, um Lebenszeichen zu messen, und der Monitor-Fortschritt, ist der einzige Weg, wie Sensordaten verwendet werden können, um die Gesundheit der Patienten zu verbessern. Dadurch werden sowohl Bürobesuche, als auch Krankenhausbesuche reduziert.
- Produktionsgesellschaften setzen Sensoren in ihren Produkten ein, um einen stream der Telemetrie zurückzugeben. In der Automobilindustrie liefern Systeme wie General Motor's OnStar, Kommunikationen, Sicherheit und Navigationsdienstleistungen. Vielleicht wichtiger offenbaren diese Telemetrie, Gebrauchsmuster, Misserfolgsraten, und andere Gelegenheiten für die Produktverbesserung, die Entwicklung und Zusammenbaukosten zu reduzieren.
- Die Proliferation von Smartphone und anderen GPS-Geräten, bietet Inserenten eine Möglichkeit an, Verbraucher ins Visier zu nehmen, wenn sie in nächster Nähe in einem Laden, einem Kaffee oder in einem Restaurant sind. Dies öffnet neue Einnahmen für Dienstleister, und bietet vielen Geschäften eine Chance an, neue Kunden ins Visier zu nehmen.
- Gewöhnlich wissen Einzelhändler, wer ihre Produkte kauft. Der Gebrauch von Sozialen Medien, und Web log Dateien von ihren ecommerce Seiten, kann ihnen helfen zu verstehen, wer ihre Produkte nicht gekauft hat, und vor allem warum nicht. Dies kann eine viel wirksamere „Mikrokundensegmentation" ermöglichen, und den Supply Chain, durch eine genauere Nachfrageplanung verbessern.
- Im Controlling kann Big Data bei der Betrug Prävention und dem Erkennen sonstiger Unternehmens-Risiken helfen
- Schließlich würden Social Media- Seiten, wie Facebook, LinkedIn oder Twitter ohne Big Data nicht existieren. Ihr Geschäftsmodell

> verlangt eine personifizierte Erfahrung im Web, das nur durch das Erfassen, und das Verwenden aller verfügbaren Daten über einen Benutzer, oder Mitglied, geliefert werden kann.[16]

So umfasst Big Data jegliche Bereiche von Dienstleistungen und ermöglicht neue Einsatzmöglichkeiten.

2.4 Probleme und Herausforderungen von Big Data

Trotz der Vorteile, die Big Data mit sich bringt, hat es auch seine Probleme und Herausforderungen. Unternehmen müssen in neue IT-Systeme investieren. Zudem kommt noch, dass es keine Standards, und wenig fertige Lösungen gibt. Es ist auch ein hoher Aufwand, da neben Technischen Fragen, die Anwender auch ihre Datenorganisation, sowie die damit zusammengehörenden Prozesse einbeziehen müssen. Laut einer Studie von BARC,[17] zählen die meisten Unternehmen folgende Probleme bei Big Data-Projekten auf:

- Fehlendes Technisches Know-how
- Fehlendes Fachliches Know-how
- Fehlende überzeugende Einsatzszenarien
- Technische Probleme
- Kosten
- Datenschutz(siehe Kapitel 2.5)
- Big Data nicht für Fachanwender im Unternehmen nutzbar[18]
- IT Infrastruktur nicht flexibel und skalierbar genug
- Netzwerkkapazität und Performance nicht auf neue Datenvolumen und -Strukturen ausgelegt
- Traditionelle, relationale Datenbank- und Storage-Systeme verhindern Analysen in Echtzeit und Analysen großer, unstrukturierter Datenmengen

Auch die so große Menge an Informationen, sowie deren Vielfalt und deren so kurze Aktualität stellt Unternehmen vor Herausforderungen. So sind innovative Big Data-Strategien von hoher Bedeutung, um im

[16] vgl.[WM_2013]
[17] vgl.[IBMG_2013]
[18] vgl.[IBMG_2013]

Datendschungel nicht zu versinken. Transparenz im Datenbestand, in den Datenquellen und in der Datenvielfalt herzustellen, ist sehr wichtig, um Daten überhaupt effektiv managen, validieren und analysieren zu können. Wer nicht weiß, welche Informationen in welcher Form überhaupt vorhanden sind, ist zum Scheitern verurteilt.[19]

Die Herausforderungen und Probleme, die mit Big Data auf ein Unternehmen zu kommen, werden nochmal zusammenfassend in vier Hauptkategorien unterteilt, die zum überblick und Verständnis dienen sollen:

- Eine umfassende Methode zum Gebrauch von Big Data.
  Die meisten Unternehmen sammeln ungeheure Mengen an Daten, aber haben keine umfassende Methode, die Informationen zu zentralisieren. Gemäß einer neuen Umfrage durch LogLogic(ein Software Infrastruktur Unternehmen), sagen mehr als 59% der 200 befragten security officers, dass sie ungleiche Systeme verwenden, um Daten zu sammeln, sie managen keine Datenerfassung, oder benutzen veraltete Kalkulationstabellen. Die richtigen analytischen Werkzeuge können bestimmt dabei helfen, Daten zu rationalisieren und zu verstehen, aber eine gut konzipierte Strategie, um Datenquellen von verschiedenen Speichern zu sortieren.

- Effektive Wege, Big Data in „Große Einblicke“ umzuwandeln.
  Daten versorgen Entscheidungsträger nicht mit der Art und Weise, wie sie ihre Jobs effektiv machen, oder die nächst besten Entscheidungen zu treffen, die auf Entdeckungen über Kundentendenzen oder anderen Enthüllungen über Marktbedingungen gestützt sind. Dies ist, wo die richtigen analytischen Werkzeuge erforderlich sind, um Datenwissenschaftlern zu helfen, Geschäftsführern, die so große Menge an Daten brauchbar zu machen, die in ihre Organisationen strömen. Dies schließt den Gebrauch von Datenvergegenwärtigungswerkzeugen mit ein, die verwendet werden können, Daten in den Zusammenhang zu stellen.

- Die richtigen Informationen den Entscheidungsträgern liefern.

[19] vgl.[BIT_2012] Kap.3, S.15-16

Um zu vermeiden, unter der riesigen Menge an Informationen begraben zu werden, sollten Unternehmen Analytik verwenden. Zu viele Unternehmen mangelt es an zusammenhängenden Methoden, Kundendaten und Geschäftsdaten, die in ihr Unternehmen fließen, richtig und effizient zu verwenden. Außerdem müssen kritische Daten von unbedeutenden und unnötigen Daten getrennt werden.

- Big Data Sachkenntnisse sind im knappen Vorrat.
  Es gibt bereits eine Knappheit an Datenwissenschaftlern auf dem Markt. Dies umfasst auch die Knappheit von Leuten, die wissen, wie man mit großen Mengen von Daten, und Big Data Ansätzen arbeitet. Unternehmen brauchen die richtige Mischung an Leuten, die dabei helfen, die Datenströme zu verstehen, die in ihr Unternehmen einfließen.[20]

2.5 Big Data und Datenschutz

Datenschutz ist ein Thema, was in vielerlei Hinsicht noch ungeklärt ist. Das Datenschutzrecht ist zu einer komplexen Materie geworden, die sich über verschiedene Gesetze verteilt, die teilweise veraltet sind, und bruchstückhaft ergänzt wurden. Der Grund, warum der Datenschutz bei Big Data ein Problem darstellt, ist der, dass es bei den meisten Anwendungen um Personenbezogene Daten handelt:

- Daten aus Sozialen Netzwerken
- Nutzungsdaten von Apps, oder Clickstreams von Webseiten
- Standortdaten von Handys, oder Fahrzeugen
- Telefonverbindungs-und Kommunikationsdaten
- Sensordaten von Maschinen
- Vertragsdaten zu gekauften Produkten und Dienstleistungen

Bei all diesen Aspekten von Big Data, müssen die gesetzlichen Vorgaben des Datenschutzrechts beachtet werden. Daher ist es sinnvoll, dass Datenschutzanforderungen bei Big Data Projekten frühzeitig

[20] vgl.[TIBCO_2012]

wahrgenommen werden. Nur so kann man Rechtsrisiken aus dem Weg gehen, ohne Zeit- und Kostenintensive Projektanpassungen durchführen zu müssen. Denn wenn die gesetzlichen Anforderungen ignoriert werden, können unter anderem Big Data Geschäftsmodelle, aufgrund der nicht Datenschutzkonformität, gefährdet werden. Wettbewerber könnten unter Umstände datenschutzrechtswidrige Nutzung von persönlichen, bzw. Kundendaten, abmahnen und durch einstweilige Verfügung unterbinden. Z.B. kann es sein, dass Daten keinen direkten Bezug zu einer Person haben, aber durch Verkettung mit anderen Daten, ist der Bezug zu einer Person herstellbar. Dadurch kann es sich dann wiederum um personenbezogene Daten handeln, sodass das Datenschutzrecht beachtet werden muss. Das Telemediengesetz verlangt z.B. bei Onlinedaten für Big Data Analysen (eingegebene Suchbegriffe, Surfverhalten, Login-Logout Zeiten), selbst wenn die Daten ohne Namen benutzt bzw. verwendet werden, die Kunden zu informieren und ihnen die Möglichkeit zu geben, die Auswertung ihrer Daten zu widersprechen. Doch bei der Einwilligung der Kunden müssen doch einige Punkte beachtet werden. Z.B. müssen die Kunden genauestens informiert werden, von wem, wie und wofür die Daten genutzt werden. Es ist auch nicht erlaubt, dass die Einwilligung in der Geschäftsbedingung versteckt ist. Auch Big Data Auswertungen von Standortdaten die durch GMS, GPS oder W-Lan ermittelt werden, müssen anonymisiert oder nur durch Einwilligung der Kunden benutzt werden.[21]

Das Thema Big Data wird von den Datenschutz-Aufsichtsbehörden nicht vernachlässigt. die Stellungnahme der Artikel 29 Arbeitsgruppe, hat sich in ihrem Papier, zum Grundsatz der „Zweckbindung“(siehe EU-Datenschutzverordnung), ausführlich mit dem Thema Big Data auseinandergesetzt. Das Ganze wird mit sehr hohen Strafmöglichkeiten in Artikel 79 (siehe EU-Datenschutzverordnung) gekoppelt. Auch Abschreckung ist der erklärte Zweck. Bis zu 2% des gesamten Umsatzes kann ein Bußgeld gegen jemanden betragen, der ohne, oder auch nur ohne ausreichende Rechtsgrundlage Daten verarbeitet, oder die Bedingungen für die Einwilligung nicht beachtet. Damit meint man aber nicht den lokalen Umsatz, sondern der weltweit getätigte Umsatz eines ganzen Jahres.[22]

[21] vgl.[HT_2013]
[22] vgl.[BR_2012]

Im Vergleich sind die EU Datenschutzrichtlinien weit aus „strenger“ als in den USA. In den USA ist Privatheit kein Grundrecht, sondern ein kommerzielles Gut, das man einklagen kann. Da US-Unternehmen auch bei europäischen Kunden ungehindert Daten einsammeln wollen, ohne sich wirklich an die Vorschriften zu halten, kann es in Zukunft zu Auseinandersetzungen kommen.

Durch die unterschiedlichen Rechtslegungen in den verschiedenen Gesellschaften, wird es auch schwierig, gegen Rechtswidrigkeiten bei Datenschutz und Datensicherheit anzugehen. Jeder beruft sich auf seine eigene Rechtslegung, was aber dazu führt, dass der Datenschutz in vielerlei Hinsicht, von den europäischen Unternehmen ignoriert wird.

Im Umkehrschluss aber, gäbe es ohne Big Data kein Google Streetview oder eine automatische Gesichtserkennung bei Sozialen Netzwerken. In Kalifornien gibt es Firmen wie 23andme.com, die DNA-Analysen für 99 Dollar anbieten. So soll man Gesundheitsrisiken besser überblicken, und genetisch entfernte Verwandte finden können. Doch wenn man seine Speichelprobe an solche Firmen abgibt, muss mit der Ungewissheit leben, was langfristig mit seinem Erbgut gemacht wird.[23]

[23] vgl.[HS_2013] Kap.6, S.34

3 Big Data Analytik

In diesem Kapitel werden die analytischen Aspekte bearbeitet. Da Big Data nicht alles neu erfindet, sondern technologisch gezielt da eingesetzt wird, wo die schon vorhandenen, etablierten Technologien, für die Big Data-Ansätze nicht ausreichen. Daher wird in Kapitel 3.1 erst mal eine Einordnung von Big Data in die Entwicklungslinien der Technologien im Unternehmen gemacht. In Kapitel 3.2 wird auf die Datenverarbeitung bzw. Analytik im Alten Modell eingegangen, um allgemein klarzustellen, wo sich die traditionelle IT-Infrastruktur befindet. In Kapitel 3.3 wird dieser Aspekt spezifiziert und es wird auf die beiden Analytischen Systeme, Business Intelligence (BI) und Data Warehouse eingegangen (DWH). In Kapitel 3.4 wird das Dokumentmanagementsystem (DMS) des Unternehmens aufgegriffen und mit Big Data verglichen. In Kapitel 3.5 wird auf die Auswirkungen von Big Data auf die Softwareentwicklung, eingegangen, und ein Vergleich gemacht, was sich mit Big Data in der Softwareentwicklung verändert hat. Das Kapitel 3 soll als Übergang für die in Kapitel 4 und 5 bearbeiteten Big Data Technologien (Hadoop, NoSQL-DBS etc.) dienen, und um zu verstehen zu welchen Zwecken diese entwickelt wurden.

3.1 Big Data- Einordnung

Es gilt für Unternehmen, die mit Big Data verbundenen Prozesse, Technologien und Qualifikationen, sowie die daraus entstehenden Herausforderungen, zu bewältigen, ohne die bereits bewährten Systeme und Prozesse zu gefährden oder vollständig neu zu entwickeln. Z.B. gehören Transaktionale Systeme, Data Warehouse, Business Intelligence, Dokumenten-Management- und Enterprise-Content-Management- Systeme zu den bereits vorhandenen Technologien.

Zukünftig wird eine Kombination von konventionellen und neuen Technologien zum Einsatz kommen, um Big Data Lösungen zu gewährleisten. Da Big Data nicht alles neu erfindet, und es vorher schon Technologien gegeben hat, um Daten aufzubereiten und für das Unternehmen bereitzustellen, sollen die bereits in vielen Unternehmen vorhanden Technologien einen Übergang auf eine Big Data- Lösung oder eine Integration mit Big Data Techniken erlauben.

Somit kann Big Data je nachdem als eine Weiterentwicklung schon bereits bestehender Technologien gesehen werden, oder als eine direkte Konkurrenz zu den etablierten Systemen, oder als eine sinnvolle Ergänzung einer Anwendungslandschaft des Unternehmens.[24,25]

3.2 Datenverarbeitung und Analytik: Das alte Modell

Traditionell, folgt die Datenverarbeitung, die zu analytischen Zwecken dient, einem ziemlich statischen Entwurf. Und zwar belaufen sich bescheidene, bis große Mengen (hängt von der Größe des Konzerns ab) an strukturierten Daten, mit stabilen Datenmodellen, durch den regulären Geschäftsablauf, über Unternehmensanwendungen, wie CRM (Customer Relationship Management), ERP (Enterprise Ressource Planning) und Finanzsysteme.

Datenintegrationswerkzeuge werden benutzt, um Daten aus Unternehmensanwendungen und transaktionalen Datenbanken, wo Datenqualität und Datennormalisierung vorkommen, und Daten in ordentlichen Reihen und Tabellen modelliert werden, zu extrahieren, zu transformieren und zu laden. Anschließend werden die modellierten, gereinigten Daten, im Data Warehouse des Unternehmens gelagert. Diese Routine kommt auf einer vorgesehenen Basis- gewöhnlich täglich, oder wöchentlich, manchmal öfter vor.[26]

Die Voraussetzungen von traditionellen Unternehmensdatenmodellen, sind im Laufe der Jahre, für Anwendungen, Datenbanken und für Datenspeicher gewachsen. Auch die Kosten und die Komplexität dieser Modelle, sind auf dem Weg, den Bedarf von Big Data zu decken, gestiegen.

Die für Big Data aber entscheidenden, unstrukturierten Daten, Logfiles und Data Streams bleiben im alten Modell, meist aus den folgenden Gründen unberücksichtigt:

- Die Datenmengen, können mit den traditionellen Relationalen Datenbanken nicht verarbeitet werden, weil sie zu groß sind. Daten werden aus Kostengründen in Archive ausgelagert und stehen für Analysen nicht zur Verfügung.

[24] vgl.[BIT_2012] Kap.5, S.22
[25] vgl.[IBMC_2012] Kap 1, S.13
[26] vgl.[WM_2013]

- Die Daten sind zu unstrukturiert, um sie unverarbeitet in einer strukturierten, relationalen Datenbank zu speichern.
- Ein weiterer Grund ist, dass die Daten zu schnell erzeugt werden, um sie mit aufwendiger Vorverarbeitung zu strukturieren und anschließend in einem Datenbanksystem einzuspielen.[27]

Traditionelle Relationale Datenbanken, sind bekannt für die folgenden Eigenschaften:

- Transaktionale Unterstützung für die **ACID** Eigenschaft:
- **Atomicity** (Atomarität): Wo alle Änderungen vorgenommen werden, als wären sie eine einzige Operation.
- **Consistency** (Beständigkeit): Am Ende jeder Transaktion, ist das System in einem gültigen Zustand.
- **Isolation** (Isolation): Die Aktionen, um die Ergebnisse zu liefern, scheinen sequentiell, einer nachdem anderen gemacht worden zu sein.
- **Durability** (Dauerhaftigkeit): Alle mit dem System vorgenommenen Änderungen sind dauerhaft.
- Die Antwortzeiten erfolgen gewöhnlich im Bruchteil einer Sekunde, während Tausende von Interaktiven Benutzern bedient werden.
- Die Menge der Daten liegt im Terabyte Bereich.
- Als Hauptprogrammierungssprache benutzt sie die SQL-92 Standards.[28]

Im Großen und Ganzen sind relationale Datenbanken, nicht in der Lage die V-Kriterien von Big Data zu bearbeiten. Daher werden andere Technologien, wie z.B. NoSQL-DBMS (siehe Kapitel 4) benutzt, um diese Probleme zu lösen.

3.3 Analytische Systeme: DWH und BI

Business Intelligence und Data Warehouse wurden frühzeitig als neue Tools für die steigenden technischen Anforderungen bei der Analyse von Daten eingesetzt. Diese Tools basieren auf bestehende Transaktionale Systeme, und erlauben einen Blick auf die Daten, der direkt bei der unternehmerischen Entscheidungsfindung unterstützen soll. D.h., dass die

[27] vgl.[GK_2013] Kap.1 , S.244
[28] vgl.[ZK_2013] Kap.1, S.4

Daten aus unterschiedlichen Abteilungen extrahiert, transformiert und in einem Zentralen Datenlager, DWH abgelegt werden. Die Eigenschaft dieser Systeme besteht darin, dass sie vor allem für strukturierte Daten geeignet sind, die auf SQL-Abfragen und Tabellenstrukturen optimiert sind. In ihrem Einsatz sind sie auf wenige, leistungsfähige und auch teure Computer optimiert.[29]

Dies ist eine Blickweise, die aus der Erstellung einer Kopie, in einer verdichteten Periode, aus den Originaldaten hervorgeht. Damit die Geschwindigkeit der Auswertung erhöht wird, wird eine feine Detaillierung der Daten weggelassen. Durch die zugrundeliegenden Prozessabläufe der Softwareentwicklung und Qualitätssicherung, ist die Anpassung der Verdichtungs-Algorithmen im DWH des Unternehmens aufwändig.[30]

Bei diesen Systemen, werden Datenmengen im niedrig-stelligen Terrabyte-Bereich verarbeitet. Diese Größenordnung findet sich auch bei kleinen Big Data-Ansätzen wieder. Dadurch überschneiden sich auch analytische Systeme und Big Data, in ihrer Aufgabenstellung. Dadurch, dass aber die Schwerpunkte beider unterschiedlich gesetzt sind, wirkt sich das bei den genutzten technologischen und organisatorischen Ansätzen aus. Daher ist ein gezielter Einsatz von Big Data-Techniken sehr wichtig, um bestehende Business-Intelligence-Lösungen zu ergänzen. Dabei kann die Auswertung großer Bestandsdaten beschleunigt werden, oder die Aktualisierungshäufigkeit von Bewegungsdaten drastisch erhöht werden.[31] Um die Schwerpunkte der analytischen Systeme und Big Data zu veranschaulichen, wird die folgende Tabelle zum Vergleich angebracht:

Analytische Systeme	Big Data
Zentrale Datenhaltung, alle Daten müssen exakt zueinander passen	Daten existieren an mehreren Stellen, Ungenauigkeiten sind akzeptabel
Qualitativ hochwertige Daten	Einfachheit der Nutzung von Daten
Strukturierte, bereinigte Daten	Daten mit unterschiedlichen Formaten
Wiederkehrende Berichte	Interaktion in Echtzeit
Periodische Erstellung	Optimiert für Flexibilität
Zentralistische Organisation	Heterogene, dezentrale Organisation

Tabelle 1: Analytische Systeme und Big Data – Vergleich der Schwerpunkte[32]

[29] vgl.[DK_2013]
[30] vgl.[BIT_2012] Kap.5, S.24
[31] vgl.[BIT_2012] Kap.5, S.24
[32] vgl.[BIT_2012] Kap.5, S.24

3.4 Big Data und Dokumentenmanagement

Für die Verwaltung elektronischer Dokumente und freier digitaler Formate, die in unstrukturierter, bzw. polystrukturierter Form vorliegen, gibt es im Unternehmen datenbankgestützte Systeme. Zu diesen Systemen gehören Dokumentmanagement-Systeme (DMS) und Enterprise-Content-Management-Systeme (ECM), die zur Unterstützung der unternehmensinternen Workflow dienen. [33,34] Es besteht ein wesentlicher Unterschied zwischen Big Data und dokumentgestütztem Workflow. Big Data unterstützt den rechtssicheren Umgang mit Dokumenten innerhalb eines Unternehmens nicht, sie dient meist der Analyse von flüchtigen Daten, wobei sich das DMS mit seiner rechtssicheren Verwaltung von Dokumenten auszeichnet (siehe Tabelle2). Um die Schwerpunkte beider zu veranschaulichen, wird die folgende Tabelle zum Vergleich angebracht:

DMS	Big Data
Rechtssichere Verwaltung von Dokumenten, z.B. Versionierung	Verarbeitung flüchtiger Daten
Einordnung in Workflow- Prozesse	Analyse von Datenmengen
Strukturierte Metadaten zur Beschreibung elektronischer Inhalte	Integration verschiedener und häufig wechselnder Formate
Für unternehmensinternen Einsatz und einem beschränktem Nutzerkreis	Prinzipiell unbeschränkter Nutzerkreis vorstellbar
Zugriff auf einzelne Dokumente aus einem großen Bestand	Auswertung großer Datenbestände mit unterschiedlichen Datenformaten
Auswertung und Suchen auf Basis von Textindizes oder Metadaten	Anpassbar an sehr unterschiedliche Formate

Tabelle2: DMS und Big Data – Vergleich der Schwerpunkte[35]

Die durch die Spartphones erzeugten Dokumente, Ton-, Bild- und Videoaufzeichnungen, sorgen für ein riesen Datenaufkommen. Da Systeme, wie die DMS oder EMC nicht ausreichend gerüstet für diese Daten sind, besteht die Herausforderung von Big Data darin, diese Datenformate zugänglich und für geschäftliche Entscheidungen und Prozesse nutzbar zu machen. Daher liegt die Konzentration vieler Big Data-Lösungen beim Zugriff auf unstrukturierte Daten, sowie deren Transformation.

Steigende Datenvolumina und verstärkt unstrukturierte Daten gehören zu den Eigenschaften von Big Data, dazu zählt auch Die Erfassung, Analyse und Integration von Daten mit hoher Geschwindigkeit. Somit wird von Big

[33] vgl.[IBMG_2013]
[34] vgl. [DAT_2013]
[35] vgl.[BIT_2012] Kap.5, S.25

Data, Anforderungen an bereits etablierte Systeme gestellt. Sie eröffnet Einsatzmöglichkeiten von neuen Technologien, die auf Big Data ausgerichtet sind.[36]

3.5 Big Data- Auswirkungen auf die Softwareentwicklung

In der Herangehensweise an Problemlösungen und in der Softwareentwicklung, gibt es auch wesentliche Unterschiede. Das Lösungsdesign wurde bislang von Geschäftsanforderungen gesteuert. Alle Anforderungen, die mit Hilfe der Analyse beantwortet werden sollen, werden vom Fachbereich festgelegt. Anschließend wird anhand der Vorgaben von der IT-Abteilung, eine Lösung mit der geforderten Funktionalität und einer festen Struktur entworfen. Die Fachabteilung fährt immer wieder das Ergebnis mit seinen definierten Queries und Reports in den transaktionalen und analytischen Systemen. Ebenso ist ein Redesign und Rebuild dieser Lösung, im Falle neuer Anforderungen notwendig.[37]

Der Big Data-Ansatz für das Software Engineering lässt sich heute mit der Aussage beschreiben, dass mehr Daten, bessere Ergebnisse liefern als intelligentere Algorithmen auf bestehenden Daten.[38] Bei Big Data zielt die Softwareentwicklung darauf ab, mit einem offenen Ausgang möglichst viele Daten zu untersuchen bzw. zu erforschen. Der Grund dafür ist, dass der Wert der Daten im Big Data Umfeld nicht bekannt ist.

Datenquellen, die interessant und brauchbar sind, werden von der IT-Abteilung und Fachabteilung identifiziert. Zur Erkundung und Auswertung von Datenquellen, wird von der IT-Abteilung eine flexible Plattform zur Verfügung gestellt. Die Einbeziehung von Datenquellen, die nachträglich gefunden wurden, in den Analyseablauf, ist möglich und sogar wünschenswert. Anschließend können die Daten von der Fachabteilung für einen Ansatz zur Problemlösung oder zum Aufstellen einer Hypothese genutzt werden. Die erzielten Ergebnisse und die Ergebnisse aus den traditionellen Analysen können kombiniert werden, um so fundiertere Geschäftsentscheidungen treffen zu können.

[36] vgl.[GK_2013] Kap.1, S.245
[37] vgl.[BIT_2012] Kap.5, S.26
[38] vgl.[BIT_2012] Kap.5, S.26

Somit gehört auch der Einsatz moderner Prozessansätze der Software-Entwicklung und Projektsteuerung, zu den Eigenschaften von Big Data. Dabei müssen die Unternehmen, die aus den neuen Prozessen, Technologien und der dazu notwendigen Qualifikationen, entstandenen Herausforderungen bewältigen. Es gilt die die bewährten Systeme und Prozesse nicht zu gefährden oder vollständig neu zu entwickeln.[39]

Damit die Unterschiede der Analyseansätze klar werden, wird die folgende Abbildung angebracht:

Abb.1Analyseansätze für traditionelle Systeme und Big Data Systeme[40]

[39] vgl.[BIT_2012] Kap.5, S.25
[40] vgl.[BIT_2012] Kap.5, S.26

4 Big Data Technologien-NoSQL DBMS

Zunächst einmal sollte gesagt werden, dass die hier in diesem Kapitel bearbeiteten Themen, nicht vertieft werden können, und zwar aus dem Grund, dass die Zeit und der Umfang dieser Bachelorarbeit begrenzt sind. Es wird darauf acht gegeben, dass alle für NoSQL wichtigen und relevanten Aspekte angesprochen werden. In diesem Kapitel geht es hauptsächlich um NoSQL Datenbanken und deren Bedeutung für die heutige Datenbankwelt. In Kapitel 4.1 wird verdeutlicht, welche neuen Big Data Komponenten, bzw. Technologien hinzugekommen sind, sprich NoSQL und Hadoop, wobei diese kurz vorgestellt werden. In Kapitel 4.2 wird die Entstehung und Bedeutung des Begriffs NoSQL erklärt, und die historischen Hintergründe hervorgebracht. Kapitel 4.3 dient nicht nur zur einfachen Definition des Begriffs NoSQL, sondern hier werden auch die Eigenschaften erklärt, um ein besseres Verständnis über die Bedeutung von NoSQL Datenbanken zu kriegen. In Kapitel 4.4 wird NoSQL kategorisiert, und die wichtigsten Kategorien werden erklärt. In Kapitel 4.5 wird am Beispiel HBase, die Bedeutung von NOSQL für Big Data nochmal aufgegriffen, um den Sinn dieses Kapitels nochmal zu verdeutlichen.

4.1 Big Data- Komponenten

Zwei Hauptblöcke werden der IT-Infrastruktur des Unternehmens hinzugefügt, um Big Data anzupassen:

1. **NoSQL** (Not only SQL):
   - Besitzt die Fähigkeit, in Echtzeit den großen Zulauf von unstrukturierten Daten und Daten ohne Schemen, zu erfassen, zu lesen und zu aktualisieren. Zu diesen Daten gehören Klick Streams, Daten aus Sozialen Medien, Log Files, Ereignis Daten, Sensor Daten und Maschinen Daten (siehe Kapitel 2).
   - NoSQL Datenbanken werden in Bezug auf Big Data dazu verwendet, um Big Data zu erwerben und anschließend zu speichern. Sie sind gut für dynamische Datenstrukturen geeignet und sind sehr skalierbar (siehe Kapitel 4.3).
   - Die in einer NoSQL-Datenbank gespeicherten Daten sind normalerweise einer hohen Vielfalt (Variety), weil die Systeme darauf ausgerichtet sind, um einfach alle Daten zu gewinnen, ohne

diese in ein festes Schema zu kategorisieren und anschließend syntaktisch zu analysieren.[41]

2. **Hadoop**:
   - Stellt Speicher Kapazitäten durch einen verteilten, shared-nothing Dateisystem und Analyse-Fähigkeit durch MapReduce zur Verfügung.
   - Sie ist eine neue Technologie, die ermöglicht, dass große Datenmengen organisiert und bearbeitet werden, während die Daten auf dem ursprünglichen Datenspeicher beibehalten werden.
   - Hadoop Distributed File System (HDFS) ist ein langfristiges Speichersystem für Web Logs zum Beispiel. Diese Web Logs werden in das Browserverhalten beim Laufen von MapReduce Programmen auf dem Cluster, und beim Generieren von angesammelten Ergebnissen auf demselben Cluster, umgewandelt.[42]

Auf diese beiden Technologien wird in den nachfolgenden Kapiteln detailliert eingegangen, wobei Hadoop erst in Kapitel 5 kommt.

4.2 Entstehung und Bedeutung des Begriffs NoSQL

Der Begriff NoSQL tauchte zum ersten Mal bei einer Datenbank von Carlo Strozzi auf. Obwohl seiner Datenbank immer noch ein relationales Datenbankmodell zugrunde lag, stellte er keine SQL-API zur Verfügung.[43,44] Die ersten noch heute bekannten Vorreiter der NoSQL-Systeme, entstanden mit Systemen, wie Berkley-DB, Lotus Notes und GT.M aus den 80er Jahren.

Erst mit Web 2.0 und dem Versuch, auch große Datenmengen zu verarbeiten, kam der eigentliche Start für NoSQL seit dem Jahre 2000. Mit den Ansätzen wie MapReduce und dem Big Table-Datenbanksystem (2004), kann man Google als den Vorreiter von NoSQL bezeichnen. Firmen wie Amazon, Yahoo und Soziale Netzwerke, wie Facebook, Twitter, MySpace etc. zogen hinterher nach.

[41] vgl.[WM_2013]
[42] vgl.[WM_2013]
[43] vgl.[EFHB_2010] Kap.1, S.1
[44] vgl.[HP_2010] Kap.2, S.1

Die heutigen klassischen NoSQL-Systeme, wie Hpertable, MongoDB, Cassandra, Voldemort, CouchDB, Redis, Riak und viele mehr, entstanden von Jahre 2006-2009.

Der heutige Begriff aber tauchte Mai 2009 in einem Weblog von Eric Evans auf, als das Team von Johan Oskarsson einen Begriff für einen Event, welches sich mit „Distributed Data Storage"-Systemen beschäftigte, suchte.[45]

Als neue Bedeutung von NoSQL, schlug im Herbst 2009 Emil Efrem, „Not only SQL" vor, was heute weitestgehend akzeptiert wird.[46]

Alles in allem kann gesagt werden, dass es Datenbanken, die stark vom relationalen Datenbankschema abweichen, schon seit Jahrzehnten gibt. Allerdings bildete sich eine Formation dieser NoSQL-Systeme als Kontrast zu dem RDBMS-Monopol erst seit 2009.

4.3 NoSQL-DBMS Definition und Eigenschaften

Zunächst einmal sollte gesagt werden, dass es weder Gremien noch Organisationen gibt, die sich wirklich um eine einheitliche Begriffserklärung bemüht haben. Daher wird hier bei der Definition, die meist benutzten Erklärungen genommen, um so eine Vereinheitlichung des Begriffs NoSQL zu erlangen.

NoSQL als eine neue Generation von Datenbanksystemen und als Kontrast gegen relationalen Datenbanken, wird durch die folgenden Aspekte definiert, die in fast allen Quellen aufzufinden sind, die den Begriff definieren:

- 1. Das Datenmodell folgt kein relationales Schema, also ist nicht relational.
- 2. Das System ist Schema frei.
- 3. ist auf eine verteilte und horizontale Skalierbarkeit ausgerichtet.
- 4. Das NoSQL-System ist Open Source (z.B. HBASE, Cassandra).
- 5. Einfache Datenreplikation aufgrund der verteilten Architektur.
- 6. Das System bietet eine einfache API
- 7. Dem System liegt meistens ein anderes Konsistenzmodell vor, wie das Eventually Consistent und BASE (Basically Available, Soft

[45] vgl.[EFHB_2010] Kap.1, S.2
[46] vgl.[HP_2010] Kap.1, S.1

State, Eventually Consistent). Demnach aber kein ACID (vgl. Kap.3.2), wie bei relationalen Datenbanken.[47,48,49]

Die oben genannten Aspekte gehen aus der Erkenntnis heraus, dass es immer schwerer wurde, die ungeheuren Datenmengen des Web 2.0-Zeilalters im Terabyte-oder sogar Petabyte-Bereich mir relationalen Datenbanken zu realisieren. Aber Punkt eins soll nicht bedeuten, dass relationale Datenbanken nicht brauchbar sind, sondern vielmehr, dass das relationale Datenmodell nicht immer das perfekte Modell ist, wie z.B. bei Big Data und ihren ungeheuren Datenmengen.

Die im Punkt zwei erwähnte *Schemafreiheit*, bezieht sich darauf, dass NoSQL-Systeme die Verantwortung im Umgang mit dem Schema, auf die Anwendung übertragen. Anders als bei relationalen Datenbanken, wo Schemaerweiterungen mit „ALTER TABLE“ durchgeführt wurden, und es keine Seltenheit war, dass darüber liegende Systeme bzw. Portale für einige Stunden lahmgelegt wurden.[50]

Die in Punkt drei angesprochene *Skalierbarkeit*, bedeutet, dass das System die Fähigkeit besitzt, den Datendurchlauf, durch Ergänzung von Ressourcen zu erhöhen, um die Lastzunahme zu adressieren. Bei der horizontalen Skalierbarkeit (Scale-out), werden die zu persistierenden Daten, horizontal über verschiedene Knoten partitioniert. D.h., dass sogenannte Knoten eingefügt werden, die dynamisch eingebunden werden, um einen Teil der Daten tragen zu können.[51]

Bei punkt fünf geht es um die Forderung einer *einfachen Replikationsunterstützung*, die aus einer logischen Konsequenz des verteilten Designs der meisten NoSQL-Anwendungen hervorgeht.

Bei Punkt sechs geht es um die Forderung einer *einfachen API*. Obwohl SQL zu den reifsten Standards der Datenbankwelt gehört, versuchen NoSQL-Datenbanken neue Wege zu gehen, auch wenn die Abfragen nicht mehr join-intensiv sind, wie das bisher der Fall war. Ein Gutes Beispiel wäre die CouchDB, hier werden die Datenbankabfragen, als REST-

[47] vgl.[TS_2011] Kap.1, S.4
[48] vgl.[EFHB_2010] Kap.1, S.2
[49] vgl.[HP_2010] Kap.2, S.1-2
[50] vgl.[EFHB_2010] Kap.1, S.3-4
[51] vgl.[DM_2012] Kap.1, S.2

Anfragen formuliert. Im Vergleich zu SQL, sind NoSQL APIs wesentlich einfacher, bietet aber weniger Funktionalität bei Abfragen an.

Beim letzten Punkt muss darauf geachtet werden, welches Konsistenzmodell das NoSQL-System benutzt. An dieser Stelle ist es angebracht das CAP-Theorem anzubringen, welches eine hohe Bedeutung für das Verständnis der Konsistenzmodelle für NoSQL-Systeme hat. Alle Arten von verteilten Systemen unterliegen dem Cap-Theorem:

- **Konsistenz (Consistency):** Nach Abschluss einer Transaktion, erreicht die verteilte Datenbank einen konsistenten Zustand. D.h., dass bei einer Transaktion der Eintrag auf einem Knoten in einer Tabelle verändert wird, und alle Lesezugriffe, denselben Wert zurückliefern, egal über welchen Knoten.
- **Verfügbarkeit (Availability):** Das System ist jederzeit verfügbar, wenn es gebraucht wird. Sie bezeichnet aber auch eine akzeptable Reaktionszeit, wobei die Akzeptanz von Anwendung zu Anwendung unterschiedlich ist. Es gibt viele e-commerce Anwendungen, wie Amazon, bei denen die Reaktionszeit immens wichtig für die Geschäftsentwicklung und den Umsatz ist.
- **Ausfalltoleranz (Partition Tolerance):** Eigenschaft einer verteilten Datenbank, Ausfälle abzufangen. D.h., wenn ein Knoten oder die Kommunikationsverbindung zwischen den Knoten ausfällt, kann das System trotzdem auf Anfragen von außen reagieren. [52,53]

In der Praxis aber können nicht alle drei dieser Eigenschaften erreicht werden, sondern nur zwei. D.h., wenn z.B. für eine hohe Verfügbarkeit und einer Ausfalltoleranz entschieden wird, so müssen die Anforderungen an die Konsistenz gelockert werden, dies besagt die Kernaussage von Brewser CAP-Theorem.

Neben dem CAP-Theorem gibt es das BASE Modell. Dieses Modell wird als Lösung zum Konflikt des CAP-Theorem herangezogen. Bei BASE wird die Konsistenz der Verfügbarkeit untergeordnet. Es ist als Gegenpart zu dem ACID Modell relationaler Datenbanken entstanden. Anders als bei dem ACID Modell, dreht sich bei BASE alles um die Verfügbarkeit. Die Konsistenz wird viel mehr als ein Übergangsprozess gesehen, und nicht als

[52] vgl.[EFHB_2010] Kap.2, S.31-33
[53] vgl.[DM_2012] Kap.1, S.2-3

einen festen Zustand nach einer Transaktion. Somit wird sie nicht unmittelbar nach jeder Transaktion, sondern erst nach einer unbestimmten Zeit der Inkonsistenz erreicht. Dabei kann die Zeit der Inkonsistenz von mehreren Faktoren, wie die durchschnittliche Reaktionszeit, Anzahl der replizierenden Knoten und dem durchschnittlichem Last des Systems abhängen. Diese Art der Konsistenz wird als Eventually Concsistency bezeichnet. Hier sind besonders die Systeme betroffen, die eine Vielzahl von replizierenden Knoten beinhalten.[54]

Dieses Thema kann auf Grund der Begrenzung der Umfang der Bachelorarbeit, nicht weiter vertieft werden. Weitere Quellen zu diesem Thema sind im Verzeichnis der *Weiterführenden Quellen* ([BT_2011-2013], [FDB_2013], [RD_2013], [SIT_2012-2014], [VW_2007]) zu finden.

Alles in allem muss gesagt werden, dass NoSQL Datenbanken ein schwächeres Konsistenzmodell verfolgen, als Relationale Datenbanken. Jeder Entwickler muss seinen eigenen Weg finden, um mit dieser Situation umzugehen, oder sich an Beispiele wie Amazon orientieren, die bereits ein eingespieltes System haben, welches seit Jahren funktioniert.

4.4 NoSQL-Kategorisierung

Immer mehr nichtrelationale Datenbanken versuchen sich unter NoSQL-Systemen zu kategorisieren. Auch große Firmen, wie Oracle oder IBM, die an der Spitze von Herstellung relationaler Datenbanken sind, versuchen einiger ihrer Produkte unter der Obhut von NoSQL-Systemen arbeiten zu lassen.

Aufgrund dieser Tatsache, musste ja auch eine Kategorisierung der NoSQL-Systeme geben, wobei NoSQL-Systeme in zwei Hauptgruppen unterteilt werden. Einmal die Kategorisierung in Kern-NoSQL-Systeme (Core) und zweitens in nachgelagerte NoSQL-Systeme (Soft). Wichtig ist es zu wissen, dass die Grenze zwischen den NoSQL-Systemen nicht eindeutig ist, und jeder die Grenzen unterschiedlich setzt.

Folgende NoSQL-Systeme werden unter diesen beiden Hauptgruppen kategorisiert:

[54] vgl.[TS_2011] Kap.9, S.181

- **NoSQL-Kernsysteme:**
    - Wide Column Stores/Column Families
    - Document Stores
    - Key/Value/Tuple Stores
    - Graphdatenbanken

- **Nachgelagerte NoSQL-Systeme:**
    - Objektdatenbanken
    - XML-Datenbanken
    - Grid-Datenbanken
    - nicht relationale Systeme…

Die vier wichtigsten Konzepte aber sind die Wide Column Stores/Column Families, Document Stores, Key/Value/Tuple Stores und Graphdatenbanken.[55,56]

Als nachgelagerte No-SQL-Systeme, werden diejenigen Datenbanken deswegen so bezeichnet, weil sie Alternativen zu relationalen Datenbanken anbieten. Dazu gehören z.B. XML-Datenbanken, weil sie ein nicht relationales Datenbankmodell, sondern die Realisation der Skalierung, verfolgen. Oder Objektdatenbanken, die erste ernstzunehmende Alternative, seit den 90er Jahren für relationale Datenbanken darstellen.

Dieses Thema wird auf Grund des Umfangs der Bachelorarbeit, und aus zeitlichen Gründen nicht weiter vertieft.

Key/Value-Systeme bieten ein einfaches System aus Schlüssel und Werten an. Die Daten bzw. Tupel werden ähnlich, wie bei relationalen Datenbanken gehandhabt. Sie werden gruppiert gespeichert, wobei hier die gruppierten Daten einem Schema folgen können, wie bei BigTable. Eine Gruppierung stellt die Aufteilung der Schlüssel in Namensräume und Datenbanken dar. Auf diese Tupel wird über einen Schlüssel zugegriffen, sodass die eindeutige Identifizierung des Tupels gewehrleistet ist. Hier sind Values nicht nur Zeichenketten, sondern Sets, Hashes oder Listen können auch Values sein.

[55] vgl.[HP_2010] Kap.4, S.6
[56] vgl.[EFHB_2010] Kap.1, S.6

Der Vorteil der Key/Value-Systeme besteht im einfachen Datenmodell, die durch das einfach gehaltene Schema entsteht, wodurch eine effizientere Datenverwaltung zu Stande kommt. Was ein Nachteil darstellt, ist die Abfragemächtigkeit, eigene Queries können oft nicht selbst geschrieben werden, daher wird sich auf die Mächtigkeit der API verlassen. Systeme, wie Amazons Dynamo, oder Redis, Voldemort und Scalaris, siehe Weiterführende Quelle ([AWS_2013], [AN_2012], [CIT_2013], [GH_2013]), sind bekannt für Key/Value-Systeme.[57]

Document-Stores haben im Gegensatz zu relationalen Datenbanken, keine Tabellen mit festen Schemata, sondern nur Dokumente. Der Begriff selbst kommt aus der Zeit von Lotus Notes, wo echte Anwenderdokumente gespeichert wurden. Ein Dokument kann sowohl eine ganze Tabelle aber auch ein gewöhnliches Tupel aus einer relationalen Datenbank sein. Die Dokumente sind strukturierte Datensammlungen, die in Formaten, wie z.B. JSON, YAML, XML oder auch in RDF ausgetauscht werden. Document-Stores legen z.B. JSON Dateien mit einer ID ab. Welches Format die ID aufweist, legt meistens die Datenbank fest. Zu den wichtigsten Document Stores, gehört die MongoDB, die Couch DB oder Riak. Die MongoDB speichert in BSON Format, welches ein Binäres Format von JSON ist und die CouchDB und Riak speichern in JSON Format. Hier werden meist einfache, grundlegende Abfragemöglichkeiten angeboten, bei denen z.B. die Möglichkeit fehlt, joins zu realisieren.[58,59]

Bei ***Column-Family-Systemen,*** ähneln die Datenstrukturen Excel-Tabellen, obwohl große Unterschiede vorliegen. Einer der Hauptunterschiede besteht darin, dass beliebige Schlüssel auf beliebig viele Key/Value Paare angewendet werden können.

Der wesentliche Unterschied zum Key/Value System besteht darin, dass diese Spalten mit beliebig vielen Key/Value Paaren erweitert werden können, d.h., dass eine Spalte mehrere Key/Value Paare als Wert haben kann. Daraus lässt sich auch der Name Column Family ableiten. Hier werden Spalten mit dem gleichen oder ähnlichen Inhalt zusammen gruppiert. Es gibt auch die Möglichkeit von Super-Columns in Form von Sub-Listen, die von einigen Systemen angeboten werden. Dabei können

[57] vgl.[TS_2011] Kap.1, S.14
[58] vgl.[EFHB_2010] Kap. 1, S.8
[59] vgl.[HP_2010] Kap.4, S. 6-7

die Keys- und Value-Listen nochmals in Form von Keyspaces oder Clustern organisiert werden.[60]

Die Column- orientierte Speicherung der Daten, kann ihre optimierte Anwendung auch in OLAP (On-line Transactional Processing) finden. Column Stores sind besser geeignet für OLAP. Aggregatsfunktionen arbeiten direkt auf Columns, so werden nur relevante Daten gelesen und Aggregationen werden viel effektiver bearbeitet.[61]

Die ***Graphdatenbank*** zählt zu der neusten und letzten Kategorie im Bereich NoSQL. Anders als bei relationalen Datenbanken, wo Daten in Tabellen gespeichert werden, verwenden Graphdatenbanken Knoten und Kanten, die einen Graphen bilden. Graphen wiederum dienen zur Repräsentation von Daten. Die Knoten repräsentieren Objekte, die Tupel aus dem relationalem Schema, und die Kanten die Beziehung untereinander. Gewöhnlich gibt es kein Schema für Graphdatenbanken, da in einem Graphen beliebig viele Knoten miteinander verbunden werden können. Sie wird meistens dann eingesetzt, wenn in RDBM-Systemen selbst referenzierende Entitäten oder Baumstrukturen vorzufinden sind. An dieser Stelle sind Freundschaftsbeziehungen in Sozialen Netzwerken ein gutes Beispiel dafür. Big Data und Neo4j wären ein Vertreter dieser Kategorie.[62]

Dadurch, dass es verschiedene Graphen gibt, gibt es auch verschiedene Graphdatenbanken. Die meiste Aufmerksamkeit bekommen zurzeit native Graphdatenbanken, die sogenannte Property-Graphen modellieren. Hier können die Knoten und Kanten mit Eigenschaften, also Properties versehen werden. Ein wesentlicher Vorteil der Graphdatenbanken gegenüber relationaler Datenbanken, ist die möglichst einfache und effiziente Traversierung, und die Spezialisierung auf vernetzte Informationen, die sich mit relationalen Datenbanken nur sehr mühselig abbilden lassen.

Die in diesem Kapitel vorgestellten Kategorien der NoSQL-Systeme, und die relationale Datenbank, werden in einer Tabelle mit ihren Eigenschaften abschließend aufgelistet:

[60] vgl.[EFHB_2010] Kap.1, S.7
[61] vgl.[SPD_2013]
[62] vgl.[DM_2012] Kap.2, S.6

Relationale DB	Server	Database	Table	Primary Key			
Key Value DB	Cluster	Keyspace		Key	Value		
Column Family DB	Cluster	Table/Keyspace	Column Family	Key	Column Name	Column Value	Super Column optional
Document DB	Cluster	Docspace		Doc Name	Doc Content		
Graph DB	Server	Graphspace	Nodes & Links				

Tabelle3: Auflistung der Eigenschaften der verschiedenen Kategorien[63]

4.5 NoSQL Datenbanken am Beispiel- HBase

HBase ist eine verteilte, nichtrelationale Datenbank, die HDFS (Hadoop Distributed File System) als Dateisystem, für die dauerhafte Abspeicherung für Big Data Projekte nutzt. Sie ist vor allem ein Column-orientiertes Datenbanksystem, zur verteilten Speicherung großer Mengen Semistrukturierter Daten. Vor allem stellt HBase in Echtzeit Lese/Schreib Zugriffe für Big Data zur Verfügung. Um ungeheure Datenmenge richten zu können, stellt sie auch eine sehr hohe Flexibilität zur Verfügung, und ist hoch konfigurierbar.[64]

Sie ist ein Beispiel für NoSQL-Datenbanken, die ein proprietäres System nach modelliert, denn sie beruht auf Google Big Table. HBase ist Open Source und ein Teilprojekt von Apache Hadoop, einem Projekt der Apache Software Foundation. Das System ist in Java implementiert. Auf die Mehrzahl der Features von relationalen Datenbanken verzichtet HBase, mit der Absicht möglichst einfach auf preisgünstiger Standarthardware zu skalieren. Sie findet in großen Firmen wie Adobe, Yahoo etc. ihren Einsatz.[65]

HBase ist eine Columnartige Datenbank, wo alle Daten in Tabellen, die aus Zeilen und Spalten bestehen, und nicht schemafrei sind, ähnlich wie bei relationalen Datenbanken gespeichert werden. Sie ist ein Vertreter der Column-Family-Systeme. Dabei steht eine Spalte für ein Attribut, und eine Zeile für einen Datensatz. Die Überschneidung von Zeilen und Spalten

[63] vgl.[EFHB_2010] Kap.1, S.9
[64] vgl.[JW_2013]
[65] vgl.[ASF_2013]

bezeichnet man als eine Zelle. Jede Zeile wird über einen eindeutigen Schlüssel identifiziert, wobei der Zugriff auf eine Tabelle über diesen Schlüssel erfolgt. Dieser kann ein errechneter Wert, ein String oder eine andere Datenstruktur sein, wobei HBase Byte-Arrays benutzt.[66]

Ein wichtiger Unterschied zwischen HBase Tabellen und relationalen Datenbanken besteht auch darin, dass die Zellen einer Tabelle und die Ausprägung einer Spalte in einer Zeile, versioniert werden. Hier wird das Ziel verfolgt, einen Teil der Verantwortung in Umgang mit dem Schema, auf die Anwendung zu übertragen (siehe Kapitel 4.3). Jeder Zellwert umfasst ein „Versions“ Attribut, das nichts anderes als ein Zeitstempel ist, bei der die Zeile eindeutig identifiziert wird. Eine Versionierung verfolgt Änderungen in der Zelle, und macht es möglich, jede Version des Inhalts abzurufen, wenn es notwendig ist. HBase speichert in absteigender Reihenfolge Daten in einer Zelle, in dem sie ein Zeitstempel benutzt, um so beim Lesen immer den aktuellsten Wert zu finden.

In HBase werden Columns in Colum-Families gruppiert. Die Column-Family einer Tabelle, und deren Eigenschaften werden von einem Schema festgelegt. Bevor überhaupt irgendwelche Daten gespeichert werden können, wird zu erst das Schema definiert und erschaffen. Trotzdem können Tabellen verändert und Spaltenfamilien hinzugefügt werden, nachdem die Datenbank arbeitet. Diese Erweiterbarkeit ist äußerst wichtig wenn man mit Big Data arbeitet, weil die Vielfalt (Variety, siehe Kapitel 2.1) der Daten vorab nicht bekannt ist. Denn Schemaerweiterungen in relationalen Datenbanken mit „ALTER TABLE“ können recht mühsam sein, da darüber liegende Portale für Stunden lahmgelegt werden können (siehe Kapitel 4.3)

Der Colum-Family Name wird als Präfix benutzt, um Mitglieder der Family zu erkennen. So erfolgt der Zugriff auf eine Spalte über eine Kombination aus einem Bezeichner der Spalte, und dem Präfix für die Colum-Family: An dieser Stelle wird ein Beispiel zum Verständnis angebracht:

```
[Column-Family] : [Column-Bezeichner]

 Bsp.:     Früchte:Banane
```

[66] vgl.[JW_2013]

In diesem Beispiel gehört die Banane zur Column-Family der Früchte.[67,68] So besteht der Column Name aus dem Colum-Family Präfix und dem Column-Bezeichner. Also ist „`Früchte:Banane`“ die Column selbst, wobei hier kein Wert zugewiesen wurde. An dieser Stelle, soll dieses Beispiel nur zum Verständnis der einfachen Syntax dienen, und wird daher nicht weiter vertieft.

Zusätzlich ist es wichtig zu wissen, dass das Präfix für die Column-Family aus druckbaren Zeichen bestehen muss, wobei das Trennzeichen nicht erlaubt ist. Stattdessen wurde dieses durch eine Konvention als „:“ festgelegt.

HBase bringt einige Vorteile mit sich. Dadurch, dass sie ein integriertes MapReduce Framework (siehe Kapitel 5.3) nutzt, hat sie durch verteiltes Rechnen von erweiterten Abfragen, die volle Leistung eines verteilten Systems. Zudem hat das HBase-Projekt eine starke, aktive Community, die von bedeutenden Organisationen unterstützt werden.

HBase sollte aber auch nur dann eingesetzt werden, wenn es wirklich um große Datenmengen geht. Bei kleinen Datenmengen ist sie eher nicht geeignet.

[67] vgl.[EFHB_2010] Kap.3, S.56
[68] vgl.[JW_2013]

5 Big Data Technologien-Hadoop

In diesem Kapitel wird die Big Data Technologie Hadoop behandelt, da sie neben anderen Technologien, wie Oracle Big Data- Appliance, IBM Infosphere, Splunk etc., zur Realisierung von Big Data dient. In diesem Kapitel werden nur die beiden Kernkomponenten HDFS und MapReduce behandelt, auf die anderen Komponenten (Apache Pig, Apache Hive etc.) wird auf Grund des begrenzten Umfangs der Bachelorarbeit nicht eingegangen. In Kapitel 5.1 wird Hadoop definiert und die Eigenschaften erläutert. Anschließend wird auf das Hadoop Framework, sprich auf HDFS in Kapitel 5.2 und auf MapReduce in Kapitel 5.3 eingegangen, da sie die Beiden Stufen bzw. Schichten des Hadoop Frameworks bilden. Das Kapitel 5.4 soll zum Verständnis der Funktionsweise von Hadoop, und der Zusammenarbeit von HDFS und MapReduce dienen. So sollen die Informationen, die in Kapitel 5.2 und in Kapitel 5.3 vermittelt wurden, nochmal bekräftigt werden. In Kapitel 5.5 werden ein paar Einsatzmöglichkeiten oder Einsatzgebiete von Hadoop in den verschiedenen Branchen aufgelistet. Hier werden potentielle Probleme dargestellt und Lösungen, die durch den Einsatz von Hadoop hervorgehen, kurz vorgestellt.

5.1 Hadoop Definition und Eigenschaften

Hadoop ist ein Software Framework, für die verteilte Verarbeitung von großen Datenbeständen, über große Cluster von Computern, bei einer Cluster-Größe von ca. 4500 Nodes. Sie ist eine open-source Implementation für Google MapReduce, und basiert auf dem einfachen Programmiermodell MapReduce. Die Hadoop Software ist ein Teil des Apache Projekts, und ist ein in Java geschriebenes Framework, für skalierbare, verteilt arbeitende Software. Das Hadoop Framework behandelt die Verarbeitung von Details und ermöglicht Entwicklern, sich fei auf die Anwendungslogik zu konzentrieren.[69,70,71]

Zudem ist Hadoop eine bekannte Technologie, die für die Realisierung von Big Data eingesetzt wird. Auch Großkonzerne, wie IBM, Intel und Oracle,

[69] vgl.[TSP_2012] Kap.3, S.38
[70] vgl.[SP_2014]
[71] vgl.[VJ_2009] Kap.1, S.4

setzen bei ihren Lösungen für Big Data, auf eine Erweiterung mit Hadoop Supports.[72] Hadoop hat die folgenden Eigenschaften:

- Bearbeitet sowohl strukturierte als auch unstrukturierte Daten
- Obgleich, mit Schema oder ohne Schema.
- Große Datenmengen, im Terabyte/Petabyte- Bereich
- und alle Arten von analytischen Aspekten.
- Ihre Kapazität ist durch Hinzufügen von zusätzlichen Clustern, erweiterbar.
- Sie nutzt die Waren (Commodity) Hardware, um so Kosten zu senken. Viele kleinere Server, ermöglichen eine Skalierung (Scale-out) in kleineren Stufen.

Wie schon erwähnt, ist Hadoop 100% Apache lizenziert, und ist Open Source:

- Es besteht kein Verkäufer Lockin
- Ist eine Community Entwicklung
- Hat ein reiches Ökosystem von zusammenhängenden Systemen wie z.B. das Speichern von Daten (HDFS), das Verarbeiten von Daten (MapReduce), NoSQL-Datenbank HBASE (siehe Kap.4.5), etc.

Sie hilft dabei, den ganzen Wert der Daten abzuleiten,

- Durch das Extrahieren der Werte aus Daten, was vorher nicht möglich war, wird der Gewinn des Unternehmens gesteigert.
- Kontrolliert die Kosten, mit kostengünstigerer Datenspeicherung, als alle anderen Plattformen.[73]

Diese Eigenschaften heißen aber nicht unbedingt, dass Hadoop immer die beste Lösung für die Verarbeitung von Daten in einem Unternehmen ist. Es hängt von mehreren Faktoren ab, die in Kapitel 5.6 erläutert werden.

5.2 Hadoop Distributed File System (HDFS)

Das Hadoop Framework besteht aus zwei Hauptschichten. Einmal dem HDFS, und zweitens dem MapReduce.

[72] vgl.[SPD_2013]
[73] vgl. [TS_2011] Kap.4, S.74

HDFS ist der erste Baustein von einem Hadoop Cluster. Sie ist ein Java-basiertes verteiltes Dateisystem, das die persistente und zuverlässige Speicherung, sowie den schnellen Zugriff auf große Datenmengen erlaubt. Es verteilt die Dateien in Blöcke und speichert sie redundant auf die Cluster. Um große Datenmengen effizient zu bearbeiten, ist es wichtig, dass die Computerwissenschaft sich in Richtung der Daten bewegt. Dazu gehört das Verwenden eines verteilten Systems, anstatt eines Hauptsystems. So wird eine einzige Datei in Blöcke gespalten, und diese Blöcke sind unter den Knoten vom Hadoop Cluster verteilt. Die Blöcke, die in HDFS benutzt werden, sind groß. Sie sind 128 MB oder größer, im Vergleich zu kleinen Blöcken mit 64MB, die mit traditionellen Dateisystemen verbunden sind.[74]

Um diese Tatsache besser nachvollziehen zu können, wird die folgende Abbildung angebracht:

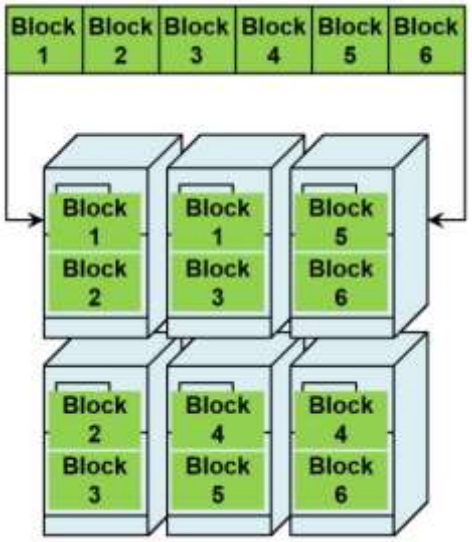

Abb.2 Distributed File System und Replication [75]

Dateien in HDFS sind “Write once” und „read many“ Dateien, d.h. es ist so ausgelegt, dass Daten idealerweise nur einmal nach HDFS geschrieben werden, und von dort vielfach ausgelesen werden. Die Eingangsdaten in HDFS werden geladen und durch das Framework von MapReduce bearbeitet, und Ergebnisse, die erzeugt werden, werden zurück im HDFS gespeichert. Die ursprünglichen Eingangsdaten werden während ihrer Existenz, in HDFS nicht modifiziert. Mit dieser Methode, beabsichtigt

[74] vgl. [SP_2014]
[75] vgl. [WM_2013]

HDFS, nicht als ein Mehrzweckdateisystem verwendet zu werden. Um die Zuverlässigkeit und die Verfügbarkeit der Daten in HDFS zu erhöhen, werden die einen bestimmten Knoten zugeteilten Daten, unter den anderen Knoten wiederholt. Die default Replikation ist dreifach. Diese Replikation hilft sicherzustellen, dass die Daten den Misserfolg, oder die nicht Verfügbarkeit eines Knotens überleben.[76,77]

HDFS hat eine master-slave Architektur. Diese Architektur besteht aus dem master namens „NameNode“ und dem slave namens „DataNode“:

1. **NameNode:**
    - HDFS besteht aus einem einzigen NameNode, der ihn zum einzigen Punkt des Misserfolgs in einem Hadoop Cluster macht. Daher wird er mit einem belastbaren, hoch verfügbaren Server versorgt.
    - Er ist dafür verantwortlich, dass das Dateisystem Metadaten nachgeht.
    - Der NameNode hat eine in-memory Datenstruktur, die das komplette Dateisystem Namespace enthält, und zeichnet die Dateien auf dem Block auf. Er enthält auch Log Dateien, die die ganze Transaktion registriert.
    - Der NameNode bestimmt das Aufzeichnen von Dateien zu replizierenden Dateiblöcken, auf DataNodes.
    - Er führt Dateisystem Namespace Operationen, wie das Öffnen, Schließen und das Umbenennen von Dateien und Verzeichnissen durch.
    - Es gibt sogenannte Checksummen auf das Lesen und Schreiben von Daten. Die Checksummen werden erst dann gültig gemacht, wenn die Daten geschrieben und zurück gelesen werden.[78,79]

2. **DataNode:**
    - DataNodes werden normalerweise in Racks organisiert, wo alle Systeme mit einem Schalter verbunden sind.

[76] vgl.[TSP_2012] Kap.3, S.39
[77] vgl. [WM_2013]
[78] vgl.[VJ_2009] Kap.1, S.6
[79] vgl.[TSP_2012] Kap.3, S.39

- Wenn ein DataNode in Gang setzt, scannt er durch sein lokales Dateisystem, und erzeugt eine Liste aller HDFS Datenblöcke, die jeder dieser lokalen Dateien entspricht.
- DataNodes antworten, um Anfragen von HDFS Clients zu lesen und zu schreiben. Sie antworten auch auf Befehle, um von NameNode erhaltene Blöcke zu erschaffen, zu löschen und zu replizieren.
- Der NameNode verlässt sich so auf periodische Nachrichten von jedem DataNode. Jedes von diesen Nachrichten enthalten einen Block Report, den der NameNode gegen seine Block Funktion und andere Dateisysteme Metadaten bestätigen kann.
- Wenn der DataNode scheitert seine Nachricht zu schicken, kann der NameNode rehabilitierende Aktionen durchführen, um die Blöcke zu replizieren, die auf diesen Nodes verloren wurden.[80,81,82]

5.3 MapReduce

Das Modell von MapReduce wird von Hadoop unterstützt und ist ebenfalls Java-basiert. Sie wurde von Google als eine Methode, zur Lösung einer Klasse von Petabyte/Terabyte Größenordnung- Problemen, mit großen Clustern von günstigen Maschinen, eingeführt. Zur Lösung der so schnell wachsenden Daten und ihrer Verarbeitung, wurden neue, alternative Algorithmen, Frameworks und Datenbankmanagementsysteme entwickelt. Das MapReduce Framework dient zur verteilten und parallelen Verarbeitung großer Mengen strukturierter und unstrukturierter Daten, welche bei Hadoop typischerweise in HDFS gespeichert sind, über große Computerclustern. [83]

MapReduce ist ein Programmiermodell, um eine verteilte Berechnung an einer massiven Skala auszudrücken. MapReduce ist eine Weise, jede Anfrage in kleinere Anfragen zu spalten, die an viele kleine Server

[80] vgl. [TSP_2012] Kap.3, S.39-40
[81] vgl. [WM_2013]
[82] vgl. [VJ_2009] Kap.1, S.6
[83] vgl. [EFHB_2010] Kap.2, S.12

gesendet werden, um so einen möglichst skalierbaren Gebrauch vom CPU zu erlangen. So ist eine Skalierung in kleineren Stufen möglich (Scale-out).

Das Modell basiert auf zwei verschiedenen Stufen für eine Anwendung:

- **Map:** Eine anfängliche Aufnahme und Transformations- Stufe, in der individuelle Input Records, parallel verarbeitet werden können.
- **Reduce:** Eine Ansammlung oder Zusammenfassungs- Stufe, bei der alle zusammengehörigen Records von einer einzigen Entität verarbeitet werden.

Das Kernkonzept von MapReduce in Hadoop ist es, dass der Input (Eingangsdaten) in logische Blöcke gespalten werden kann. Jeder Block kann unabhängig am Anfang, von einer Map-Task bearbeitet werden. Die Ergebnisse von diesen individuell arbeitenden Blöcken, können physisch in verschiedene Sätze verteilt, und anschließend sortiert werden. Dann wird jeder sortierte Block an die Reduce-Task weitergeleitet.

Ein Map-Task kann auf irgendwelchen berechneten Nodes auf dem Cluster laufen, und multiple Map-Tasks können parallel auf dem Cluster laufen. Er ist dafür verantwortlich, die Input Records in Key/Value Paare zu transformieren. Der Output (Ausgangsdaten) von allen Maps wird aufgeteilt, und jede Aufteilung wird sortiert. Es gibt aber nur eine Aufteilung für jeden Reduce-Task. Die Keys jeder sortierten Aufteilung und die Values verbunden mit den Keys, werden vom Reduce-Task verarbeitet. Die multiplen Reduce-Tasks können dann parallel auf dem Cluster laufen.[84]

Der Anwendungsentwickler muss nur vier Sachen dem Hadoop Framework zur Verfügung stellen:

- Eine Klasse, die die Input Records lesen und sie in Key/Value Paare pro Record transformieren kann.
- Eine Map Methode
- Eine Reduce Methode
- Und eine weitere Klasse, die die Key/Value Paare, die die Reduce Methode ausgibt, in Output Records transformiert.[85]

[84] vgl.[VJ_2009] Kap.1, S.1-2
[85] vgl.[TSP_2012] Kap.3, S.40-41

Auch MapReduce ist eine Master-Slave Architektur. Master ist der sogenannte JobTracker, und Slave sind die sogenannten TaskTrackers (Hunderte oder Tausende von TaskTrackern).

Der JobTracker weiß alles über vorgelegte Jobs. Als Jobs werden benutzervorgelegte Map und Reduce Implementierungen für einen Datensatz bezeichnet, die in Java geschrieben werden. Der JobTracker teilt Jobs in Tasks und entscheidet, wohin jeder Task laufen soll. Er ist in ständiger Kommunikation mit TaskTrackers.

TaskTrackers führen Tasks durch. Als Task wird ein einzelner Mapper oder Reducer Task bezeichnet. Fehlgeschlagene Tasks werden automatisch neu wiederholt. Ein TaskTracker kontrolliert die Ausführung von jedem Task, und sendet dem JobTracker ständig ein Feedback. Jeder DataNode führt einen TaskTracker aus.[86]

5.4 Funktionsweise von Hadoop

Nachdem HDFS und MapReduce vorgestellt wurden, ist es angebracht eine zusammenfassende Erklärung, für die Funktionsweise von Hadoop zu liefern. Daher werden Dinge wiederholt, die schon in den vorherigen beiden Kapiteln vorkamen. Dies soll zum besseren Verständnis von Hadoop und die Zusammenarbeit von HDFS und MapReduce dienen.

Ein typischer Hadoop Cluster besteht aus einem NameNode, der als MasterNode für das HDFS dient, aus einem Node, namens JobTracker, der Jobs erteilt und als zentralen Ort für das Vorlegen und Verfolgen von MapReduce Jobs fungiert, und aus mehreren Slave Nodes, die sowohl den DataNode als auch den TaskTracker führen. Die Daten aus HDFS sind in den DataNodes enthalten.

Die TaskTrackers führen die Map und Reduce Funktionen durch. Die Joberteilung von MapReduce, wird in Hadoop durch JobTracker durchgeführt. Die Verarbeitung von Map und Reduce wiederum, wird durch TaskTrackers durchgeführt. Gewöhnlich befindet sich der JobTracker mit dem NameNode auf dem Master und die TaskTrackers mit den DataNodes auf den Slaves. Der JobTracker teilt Map und Reduce Jobs zu den TaskTrackers ein, mit dem Bewusstsein einer Datenposition. Der JobTracker Kommuniziert mit dem NameNode, um die Position der Daten

[86] vgl.[WM_2013]

zu bestimmen. Es macht die TaskTrackers ausfindig, die in der Nähe von den Daten sind. Zudem legt es die Jobs dem Tasktracker vor, und wartet auf Nachrichten vom TaskTracker, und bestimmt und aktualisiert den Status des Jobs. Wie der NameNode, ist auch der JobTracker der einzige Punkt der zum Scheitern bzw. Misserfolg führen kann. Wenn der JobTracker versagt, werden alle restlichen Jobs blockiert bzw. gestoppt.[87,88]

Ein typischer MapReduce Job startet mit dem JobTracker. Wenn ein Job eingereicht wird, ist die Input und Output- Adresse (im HDFS) in der Konfigurationsinformation zur Verfügung gestellt. Die Input Daten sind auf einen Satz von unabhängigen Blöcken gespalten, die von sogenannten Mappern, auf eine mitwirkende Art verarbeitet werden.

MapReduce kann als ein Mapper gesehen werden, der auf alle Input Key/Value Paare angewendet wird, und der Reducer wird auf alle Values angewendet, die denselben Key haben. Um das Konzept des Hadoop Frameworks besser zu verstehen, wird die folgende Abbildung angebracht:

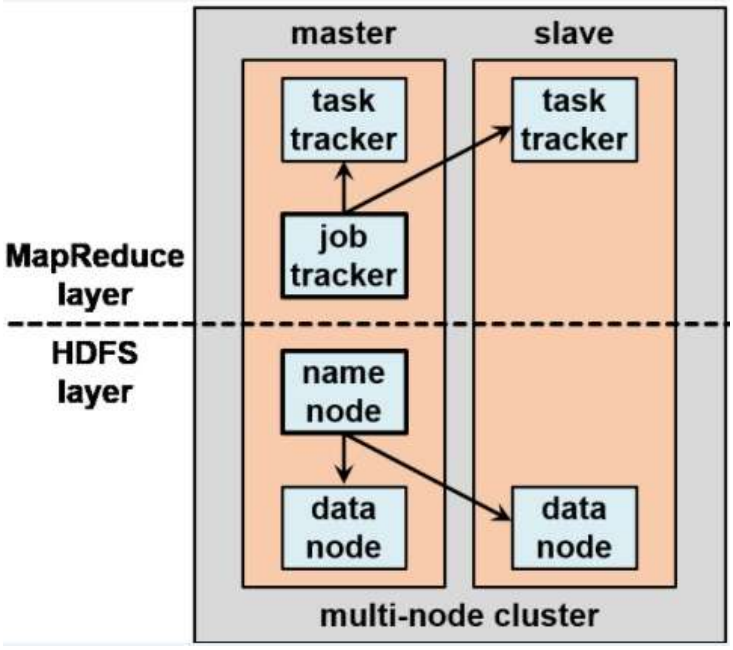

Abb.3 Hadoop Framework mit HDFS und MapReduce[89]

Der JobTracker nutzt die Kenntnis der Dateiblöcke, um zu entscheiden, wie viele TaskTrackers erstellt werden sollen. Es versucht die Tasks auf der selben Maschine zu teilen, wo sich die physischen Daten befinden, oder zuletzt auf dem selben Rack. Der JobTracker kontrolliert den Fortschritt der

[87] vgl.[TCM_2013]
[88] vgl.[TSP_2012] Kap.3, S.43
[89] vgl.[WM_2013]

MapReduce Jobs, und ist verantwortlich für das Koordinieren der Ausführung der Mapper und Reducer. Die wirkliche Verarbeitung der Daten wird durch die Map und Reduce Funktionen, die von den Benutzern geschrieben wurden, vollbracht.[90,91]

5.5 Einsatzszenarien für Hadoop

Hadoop kann in vielen Brachen eingesetzt werden. Deswegen werden einige Beispielszenarien angebracht, bei der mögliche Probleme mittels Hadoop gelöst werden können. Dies soll zum besseren Verständnis von Hadoop zu dienen:

1. **Modellieren von True Risk**
    - ***Herausforderung:*** Wie viel Risikoaussetzung hat eine Organisation wirklich mit einem Kunden? Das Analysieren von Multiplen Quellen von Daten, über multiple Brachen eines Unternehmens.
    - ***Lösung mit Hadoop:*** Beziehen und Ansammeln von ungleichen Datenquellen, wie z.B. Anruf Aufzeichnungen, Chat Sitzungen, Emails, Bank Aktivitäten.
    - ***Struktur und Analyse:*** Sentiment Analyse, Entwickeln eines Graphen, typische Muster-Erkennung.
    - ***Typische Branchen:*** Finanz-Dienstleistungen (Banken, Versicherungsgesellschaften etc.)

2. **PoS Transaktionsanalyse**
    - ***Herausforderung:*** Analyse von Point of Sale (PoS) Daten, um Promotionen in Visier zu nehmen und Operationen zu managen. Die Quellen sind komplex und der Datenumfang wächst über Ketten von Läden und anderen Quellen.
    - ***Lösung mit Hadoop:*** Eine Menge von Verarbeitungs-Frameworks (HDFS, MapReduce) erlauben die parallele Ausführung über große Datensätze.

[90] vgl. [KT_2010] Kap.2, S.5-7
[91] vgl.[TSP_2012] Kap.3, S.44

- ***Mustererkennung:*** Optimierung über multiple Datenquellen. Das Verwenden der Information, um Anforderungen bzw. Nachfrage voraus zu sagen.
- ***Typische Branche:*** Einzelhandel

3. Analysieren von Netzwerkdaten zur Vorhersage von Fehlern

- ***Herausforderung:*** Das Analysieren von Datenreihen in Echtzeit, aus einem Netzwerk von Sensoren. Mit der Zeit ist das Kalkulieren der durchschnittlichen Frequenz, auf Grund der Analyse Notwendigkeit Terrabytes von Daten, recht mühsam geworden.
- ***Lösung mit Hadoop:*** Berechnung dieser Daten durch Expandieren von einfachen Abfragen, zu komplexerem Data Mining. So hat man ein besseres Verständnis darüber, wie das Netzwerk auf Veränderungen reagiert. Getrennte Anomalien können miteinander verbunden werden.
- ***Typische Branchen:*** Telekommunikation, Datencenter, Dienstprogramme.

4. Kundenanalyse

- ***Herausforderung:*** Warum verliert ein Unternehmen Kunden? Daten zu diesen Faktoren kommen aus diversen Quellen und bilden eine Herausforderung, sie zu analysieren.
- ***Lösung mit Hadoop:*** Bauen schnell ein Verhaltensmodell aus ungleichen Datenquellen.
- ***Strukturieren und Analysieren mit Hadoop:*** Dazu zählt das Traversieren von Daten, die Erstellung eines Graphen und Mustererkennung durch verschiedene Informationen aus Kundendaten.
- ***Typische Branchen:*** Telekommunikation, Finanz Dienstleistung.[92]

Die Einsatzgebiete von Hadoop umfassen alle Bereiche, die mit großen, semistrukturierten Datenmengen zu tuen haben, sprich alle Bereiche, die Big Data umfassen.

[92] vgl.[WM_2013]

5.6 Hadoop Vorteile und Probleme

Einer der wichtigsten Vorteile von Hadoop ist es, dass Große Datenmengen im Terabyte, Petabyte Bereich analysiert werden können. Da Hadoop die Daten in Blöcke spaltet bzw. partitioniert und die Daten auf bestimmten Nodes zur Analyse zugewiesen werden, müssen die Daten nicht einheitlich sein, weil jeder Datenblock von einem separatem Prozess, auf einem anderen Node verarbeitet wird. So wird der Faktor unstrukturierter Daten von Big Data nicht zu einem Problem.

Ein weiterer Vorteil von Hadoop ist die parallele Verarbeitung der Daten, die zu einer schnelleren Analyse-Geschwindigkeit führt. Dies ist bei Big Data sehr hilfreich, da die Datenmenge ständig wächst. Sie ist dann besser nützlich, wenn sie zeitnah bzw. in Echtzeit analysiert wird. Auch wenn die Daten ständig wachsen, ist es möglich Hadoop-Cluster durch Hinzufügen zusätzlicher Cluster zu skalieren.

Da Hadoop Open-Source ist, kann Apache Hadoop kostenlos heruntergeladen werden. Dadurch ist es möglich einen recht leistungsfähigen Hadoop-Cluster zu erstellen, ohne dass dabei hohe Kosten entstehen.

Ein weiterer Vorteil besteht auch in der Replikation der Daten, die für eine Robustheit des Hadoop-Clusters sorgt. Wenn Daten zur Verarbeitung bzw. zur Analyse zu einem Node gesendet werden, werden diese auch auf andere Nodes im Cluster repliziert. Und wenn es zu einem Ausfall des Nodes kommt, so können die Daten trotzdem analysiert werden, weil zusätzliche Kopien der Daten auf anderen Nodes vorhanden sind.

Auch wenn Hadoop viele Vorteile mit sich bringt, so heißt dies nicht, dass sie immer die geeignetste Wahl für die Datenanalyse in einem Unternehmen ist. Wenn ein Unternehmen nicht mit großen Datenmengen, die im Terabyte oder Petabyte Bereich liegen, arbeitet, so macht es keinen Sinn Hadoop einzusetzen, da sie für große Datenmengen geeignet ist.[93]

Ein weiteres Problem besteht darin, wenn die Analyse für den Einsatz in einer parallelen Verarbeitungsumgebung nicht angepasst werden kann. Da

[93] vgl.[TT_2013]

Hadoop darauf basiert, dass Daten zerlegt und parallelisiert auf Nodes analysiert werden. Wenn Hadoop aber nicht in den Rest der Datenverwaltungsinfrastruktur integriert wird, wird es jedoch schnell zu einer weiteren Dateninsel, die die IT-Umgebung des Unternehmens noch komplexer macht. Daher ist es ein sehr wichtiger Aspekt, Hadoop mit anderen Datenverarbeitungs- und Analysesystemen, wie Business Intelligence etc. zu verbinden. Wenn ein Unternehmen eine bestimmte Art der Analyse braucht, ist es notwendig, dass die gewonnenen Daten außerhalb von Hadoop überführt werden können.[94]

Hadoop besteht nicht nur aus den Kernkomponenten HDFS und MapReduce. Sie hat ein reiches Ökosystem, dass aus mehreren Komponenten besteht, die im Laufe der Entwicklung von Hadoop dazugekommen sind. Um einen Überblick zu kriegen werden im Folgenden zwei dieser Komponenten aufgelistet:

- **Apache Pig**: Bildet für MapReduce-Jobs eine Abstraktionsschicht. Sie besteht aus einer abstrakten Skriptsprache, die den Datenfluss eines MapReduce-Jobs beschreibt.
- **Apache Hive**: Ist eine Abstraktionsschicht, die auf dem MapReduce Framework basiert. Ermöglicht Aufgaben, wie Aggregation, Analysen und Abfragen von Datenmengen, die in HDFS gespeichert sind.[95]

Durch diese Komponenten erhöht sich auch die Komplexität des Einsatzes von Hadoop im Unternehmen. Weil zuerst geklärt werden muss, welche dieser Komponenten in Betracht gezogen werden, und ob das Unternehmen genügend Fachleute hat, diese Technischen Aspekte umzusetzen.

Auch bringt die Einarbeitungszeit mit dem Aufbau, dem Betrieb und der Unterstützung, die für die Hadoop-Cluster notwendig sind, einen großen Nachteil mit sich. Daher darf der Einsatz von Hadoop, an Aufwand nicht unterschätzt werden, sie ist viel mehr als ein Projekt zu betrachten, dessen Umsetzung an Zeit und Geduld bedarf, und eine Herausforderung für das Unternehmen darstellt.

[94] vgl. [IC_2014]
[95] vgl. [SU_2013]

Fazit

In dieser Bachelorarbeit geht es um Big Data und ihre Technologien Hadoop und NoSQL.

Big Data ist in der Lage neue Perspektiven für das Unternehmen zu eröffnen. Aus unstrukturierten Daten, wertvolle Informationen zu gewinnen, und für die Entscheidungen im Unternehmen zu nutzen, um so Wettbewerbsvorteile gegenüber Konkurrenten zu erlangen, liegt im Bereich des machbaren, was aber nicht heißen soll, dass es keine Herausforderungen und Probleme mit sich bringt.

Da heute in einer Digitalen Welt gelebt wird, steigt die Anzahl der Datenmengen stetig. Zu 80% der Daten, die erzeugt werden sind unstrukturiert, d.h., sie weisen kein bestimmtes Format auf. Durch Faktoren, wie Soziale Medien (Facebook, Twitter, Yahoo etc.), E-Mails, Web Blogs, mobile Geräte, Sensoren und Kameras entstehen zusätzlich Terabyte, Hexabyte an Daten. Es wird vermutet, dass die Daten bis 2015 um 5 Hexabyte ansteigen werden, und bis zum Jahre 2020 um das 20 fache.

Genau an dieser Stelle setzt Big Data an. Big Data bezeichnet den Einsatz von großen Datenmengen, die aus verschiedenen Quellen gewonnen werden, um wirtschaftlichen Nutzen zu erlangen. Sie umfasst jede Art von Daten, die in Beziehung zum Geschäftsmodell des Unternehmens stehen, und die Möglichkeit, sie zu analysieren und zu verarbeiten. Sie wird durch die vier Eigenschaften Volume (Datenmenge), Velocity (Geschwindigkeit bzw. Verfügbarkeit), Variety (Vielfalt) und Veracity (Wahrhaftigkeit) beschrieben. Diese Eigenschaften, werden als die V-Kriterien von Big Data bezeichnet.

Die so große Menge an Informationen, sowie deren Vielfalt und deren so kurze Aktualität, stellt Unternehmen und ihre IT-Infrastruktur, sowie ihre Technologien vor Herausforderungen. Traditionelle, relationale Datenbank- und Storage-Systeme verhindern Analysen in Echtzeit und Analysen großer, unstrukturierter Datenmengen. Zudem können diese Daten nicht mit traditionellen relationalen Datenbanken verarbeitet werden, weil sie zu groß sind. Sie sind nicht in der Lage, die V-Kriterien von Big Data zu bearbeiten. Auch die vorhandenen Analytischen Systeme, wie Business

Intelligence und Data Warehouse, stoßen an ihre Grenzen. Diese Tools basieren auf bestehende Transaktionale Systeme, und erlauben einen Blick auf die Daten, der direkt bei der unternehmerischen Entscheidungsfindung unterstützen soll. Die Eigenschaft dieser Systeme besteht darin, dass sie vor allem für strukturierte Daten geeignet sind, die auf SQL-Abfragen und Tabellenstrukturen optimiert sind. Auch hier bleiben die unstrukturierten Daten, die für Big Data entscheidend sind außen vor.

In dieser Bachelorarbeit wird als Lösung dieser Probleme auf die Big Data-Technologien NoSQL und Hadoop gesetzt.

Big Data wird je nachdem als eine Weiterentwicklung schon bereits bestehender Technologien gesehen, oder als eine direkte Konkurrenz zu den etablierten Systemen, oder als eine sinnvolle Ergänzung einer Anwendungslandschaft des Unternehmens. Es sollte betont werden, dass sie keine „Ersetzung“ der vorhandenen Technologien ist, sondern viel mehr eine „Ergänzung“.

NoSQL besitzt die Fähigkeit, in Echtzeit den großen Zulauf von unstrukturierten Daten und Daten ohne Schemen zu erfassen, zu lesen und zu aktualisieren. Zu diesen Daten gehören Klick Streams, Daten aus Sozialen Medien, Log Files, Ereignis Daten, Sensor Daten und Maschinen Daten. NoSQL Datenbanken werden in Bezug auf Big Data dazu verwendet, um Big Data zu erwerben und anschließend zu speichern. Sie sind gut für dynamische Datenstrukturen geeignet und sind sehr skalierbar. Die in einer NoSQL-Datenbank gespeicherten Daten, sind normalerweise einer hohen Vielfalt (Variety) ausgesetzt, weil die Systeme darauf ausgerichtet sind, um einfach alle Daten zu gewinnen, ohne diese in ein festes Schema zu kategorisieren, und anschließend syntaktisch zu analysieren. Der entscheidende Unterschied zwischen dem relationalem und dem NoSQL System, besteht im Konsistenzmodell. Relationale Datenbanksysteme arbeiten nach dem ACID-Modell (Atomicity, Consistency, Isolation, Durability), wobei NoSQL-Systeme einem anderen, wie z.B. dem Eventually Consistency- Modell unterliegen.

Alle Arten von verteilten Systemen unterliegen dem CAP-Theorem (Consistency, Availability, Partition). In der Praxis aber können nicht alle drei dieser Eigenschaften erreicht werden, sondern nur zwei. D.h., wenn z.B. für eine hohe Verfügbarkeit und einer Ausfalltoleranz entschieden wird,

so müssen die Anforderungen an die Konsistenz gelockert werden, dies besagt die Kernaussage von Brewser CAP-Theorem.

Die angeführten Aspekte für NoSQL-Datenbanken sollen nicht bedeuten, dass relationale Datenbanken nicht brauchbar sind, sondern vielmehr, dass das relationale Datenmodell nicht immer das perfekte Modell ist, wie z.B. bei Big Data und ihren ungeheuren Datenmengen. Schließlich hat das relationale Datenmodell unumstrittene Vorteile, die sich in zuverlässigen Systemen seit Jahrzehnten bewehrt hat und immer noch bewehrt.

Eine weitere Technologie, auf die in dieser Bachelorarbeit gesetzt wird, ist Hadoop. Hadoop ist ein Software Framework, für die verteilte Verarbeitung von großen Datenbeständen, über große Cluster von Computern, bei einer Cluster-Größe von ca. 4500 Nodes. Sie ist eine Open-Source Implementation für Google MapReduce, und basiert auf dem einfachen Programmiermodell MapReduce. Die Hadoop Software ist ein Teil des Apache Projekts, und ist ein in Java geschriebenes Framework, für skalierbare, verteilt arbeitende Software.

Sie besteht aus den beiden Kernkomponenten HDFS und MapReduce. Sie sind für die beiden Kernaufgaben, dem Speichern und Verarbeiten großer Datenmengen zuständig.

HDFS ist der erste Baustein von einem Hadoop Cluster. Sie ist ein Java-basiertes verteiltes Dateisystem, das die persistente und zuverlässige Speicherung, sowie den schnellen Zugriff auf große Datenmengen erlaubt.

Das Modell von MapReduce wird von Hadoop unterstützt und ist ebenfalls Java-basiert. Sie wurde von Google als eine Methode, zur Lösung einer Klasse von Petabyte/Terabyte Größenordnung- Problemen, mit großen Clustern von günstigen Maschinen, eingeführt. MapReduce ist ein Programmiermodell, um eine verteilte Berechnung an einer massiven Skala auszudrücken.

Hadoop bringt einige Vorteile mit sich, wie z.B. das Verarbeiten und Analysieren von Datenmengen im Terabyte, Petabyte Bereich, oder die parallele Verarbeitung der Daten, die eine schnelle Analysemöglichkeit bieten. Jedoch bedeuten diese Vorteile nicht, dass Hadoop ohne weiteres im Unternehmen eingesetzt werden kann. Es müssen mehrere Aspekte in Betracht gezogen werden. Z.B. ob Hadoop, als Technologie eine geeignete

Wahl für das jeweilige Unternehmen ist. Wenn ein Unternehmen nicht mit großen Datenmengen, die im Terabyte oder Petabyte Bereich liegen, arbeitet, so macht es keinen Sinn Hadoop einzusetzen, da sie für große Datenmengen geeignet ist. Ein weiteres Problem besteht darin, wenn die Analyse für den Einsatz in einer parallelen Verarbeitungsumgebung nicht angepasst werden kann. Da Hadoop darauf basiert, dass Daten zerlegt und parallelisiert auf Nodes analysiert werden. Wenn Hadoop aber nicht in den Rest der Datenverwaltungsinfrastruktur integriert wird, wird es jedoch schnell zu einer weiteren Dateninsel, die die IT-Umgebung des Unternehmens noch komplexer macht. Daher ist es ein sehr wichtiger Aspekt, Hadoop mit anderen Datenverarbeitungs- und Analysesystemen, wie Business Intelligence etc. zu verbinden. Wenn ein Unternehmen eine bestimmte Art der Analyse braucht, ist es notwendig, dass die gewonnenen Daten, außerhalb von Hadoop überführt werden können.

Da diese Bachelorarbeit eine reine Literaturarbeit ist, hat meistens der Bezug zu diesen Technologien gefehlt, weil keine praktischen Erfahrungen dies bezüglich vorhanden waren. Bei Hadoop war das Problem, dass erst nachhinein bemerkt wurde, dass sie nicht als eine einzelne Technologie zu betrachten ist, sondern viel mehr als ein großes Projekt, das aus mehreren Komponenten besteht, und nicht nur aus den beiden Kernkomponenten HDFS und MapReduce. Dies bedeutet für Unternehmen wiederum, dass sie erst einmal feststellen müssen, welche Komponenten für sie in Frage kommen und welche nicht. Ob die Technischen Voraussetzungen im Unternehmen vorhanden sind und es die richtigen Fachleute gibt, die diese Technologien bzw. Komponenten in die IT-Infrastruktur anpassen können.

Viele Unternehmen sind daher zwiegespalten, da die Realisierung von Big Data, einige Herausforderungen mit sich bringt. Die IT-Infrastruktur des Unternehmens muss auf ein flexibleres und offeneres System umstrukturiert werden. Dies bedeutet für Unternehmen, dass sie umdenken, offen für neue Technologien, und vor allem bereit sein müssen, in diese neuen Technologien, wie Hadoop und NoSQL zu investieren. Daher ist auch der Kostenfaktor einer der Gründe, warum Unternehmen nicht in Big Data investieren wollen. Es ist für einige ein unsicheres Territorium. Viele wissen nicht, was sie erwartet, ob ihr Unternehmen überhaupt die Kapazität bzw. das Potential besitzt, Big Data zu realisieren.

Literaturverzeichnis

[ASF_2013] The Apache Software Foundation. (2013). *Apache HBASE*. Abgerufen am 12. Dezember 2013 von http://hbase.apache.org/

[BR_2012] Becker, R. (2012). *EU-Datenschutzverordnung*. Abgerufen am 25. Oktober 2013 von http://www.eu-datenschutzverordnung.de/

[BIT_2012] BITKOM Bundesverband Informationswirtschaft Telekommunikation und neue Medien e.V. (2012). *Big Data im Praxiseinsatz, Szenarien, Beispiele, Effekte*. Abgerufen am 5. November 2013 von http://www.bitkom.org/files/documents/BITKOM_LF_big_data_2012_online(1).pdf

[DAT_2013] DATACOM Buchverlag GmbH. (2013). *IT WISSEN , das grosse Online-Lexikon für Informationstechnologie*. Abgerufen am 12. November 2013 von DMS (document management system): http://www.itwissen.info/definition/lexikon/document-management-system-DMS-Dokumenten-Managementsystem.html

[DM_2013] David, M. (2013). *Überblick über NoSQL Datenbanken*. Abgerufen am 9. Dezember 2013 von Universität zu Lübeck: http://media.itm.uni-luebeck.de/teaching/ws2012/sem-sse/mario-david-nosql-ausarbeitung.pdf

[DK_2013] Dominik Klein, P. T.-G. (2013). *Gesellschaft für Informatik*. Abgerufen am 13. Oktober 2013 von Big Data: http://www.gi.de/nc/service/informatiklexikon/detailansicht/article/big-data.htm

[EFHB_2010] Edlich, S., Friedland, A., Hampe, J., & Brauer, B. (2010). *NoSQL Einstieg in die Welt Nichtrelationaler Web 2.0 Datenbanken.* München: Carl Hanser Verlag.

[GK_2013] Gleich, R., & Klein, A. (2013). *Der Controlling-Berater.* Freiburg: Haufe-Lexware GmbH & Co. KG.

[HP_2010]Heinze, P. (9. Juli 2010). *NoSQL-Datenbanken*. Abgerufen am 10. Dezember 2013 von Friedrich Schiller Universität Jena: http://www.informatik.uni-jena.de/dbis/lehre/ss2010/saas/material/Ausarbeitung07-Heinze.pdf

[HT_2013] Helbing, T. (2013). *Helbing Kanzlei für IT- und Datensicherheit*. Abgerufen am 25. Oktober 2013 von BIG DATA UND DATENSCHUTZRECHT - EINFÜHRUNG, ÜBERSICHT UND WEBINAR: http://www.thomashelbing.com/de/big-data-datenschutzrecht-einfuehrung-uebersicht-webinar

[HS_2013] Heuer, S. (Januar 2013). Kleine Daten, Grosse Wirkung-Big Data. Düssledorf, NRW, Deutschland.

[IBMC_2012] IBM Corportaion. (2012). *IBM Institute for Business Value*. Abgerufen am 10. November 2013 von Analytics: Big Data in der Praxis: http://www-935.ibm.com/services/de/gbs/thoughtleadership/GBE03519-DEDE-00.pdf

[IBMG_2013] IDG Business Media GmbH. (2013). *Computerwoche*. Abgerufen am 21. Oktober 2013 von Big Data: http://www.computerwoche.de/g/big-data-vorteile-und-probleme

[IC_2014] Informatica Corporation. (2014). *Hadoop*. Abgerufen am 5. Januar 2014 von http://www.informatica.com/de/vision/harnessing-big-data/hadoop/

[INT_2013] Intel Corporation. (2013). *Big-Data Analyse beginnt mit Intel*. Abgerufen am 16. Oktober 2013 von www.intel.de/bigdata

[JW_2013] John Wiley & Sons, Inc. (2013). *For Dummies Making Everything easier*. Abgerufen am 10. Dezember 2013 von Store Big Data with HBase: http://www.dummies.com/how-to/content/store-big-data-with-hbase.html

[KT_2010]König, T. (31. März 2010). *MapReduce-Konzept*. Abgerufen am 3. Januar 2014 von http://dbs.uni-leipzig.de/file/seminar_0910_koenig_ausarbeitung.pdf

[SAS_2013] SAS Institute Inc. (2013). *Competence Network Business Analytics*. Abgerufen am 20. Oktober 2013 von Big Data Analytics: http://sas-competence-network.com/business-analytics/content/big_data_analytics

[SP_2014] Speicherguide.de. (2014). *Hadoop: Was ist es und was leistet es?* Abgerufen am 3. Januar 2014 von Das Storage Magazin: http://www.speicherguide.de/management/big-data/hadoop-was-ist-es-und-was-leistet-es-15378.aspx

[SPD_2013] Störl, P. D. (Oktober 2013). *Datenbanktechnologien für Big Data*. Abgerufen am 3. Januar 2014 von https://www.fbi.h-da.de/fileadmin/personal/u.stoerl/talks/BigData-DS-Berlin-short.pdf

[SU_2013] Seiler, U. (14. August 2013). *Einführung in Hadoop- Die wichtigsten Komponenten von Hadoop*. Abgerufen am 6. Januar 2014 von https://blog.codecentric.de/2013/08/einfuhrung-in-hadoop-die-wichtigsten-komponenten-von-hadoop-teil-3-von-5/

[TCM_2013] Taggart, C. M. (2013). *Hadoop/MapReduce*. Abgerufen am 3. Januar 2014 von Object oriented Framework Presentation: http://www.cs.colorado.edu/~kena/classes/5448/s11/presentations/hadoop.pdf

[TIBCO_2012] TIBCO Spotfire. (14. März 2012). *TIBCO Spotfire's Business Intelligence Blog*. Abgerufen am 23. Oktober 2013 von The 4 Biggest Problems with Big Data: http://spotfire.tibco.com/blog/?p=10941

[TS_2011] Tiwari, S. (2011). *Professional NOSQL.* Indianapolis, Indiana: John Wiley & Sons, Inc.

[TSP_2012] Tilley, S., & Parveen, T. (2012). *Software Testing in the cloud .* Heidelberg; New York;Dordrecht;London: Springer.

[TT_2013]TechTarget. (Juli 2013). *Hadoop-Cluster: Vorteile und Herausforderungen für Big-Data-Analytik*. Abgerufen am 5. Januar 2014 von http://www.searchenterprisesoftware.de/tipp/Hadoop-Cluster-Vorteile-und-Herausforderungen-fuer-Big-Data-Analytik

[VJ_2009] Venner, J. (2009). *Pro Hadoop.* USA: Apress.

[WM_2013] Wag Mobile Inc. (2013). Big Data and Hadoop. USA, Redmond.

[ZK_2013] Zadrozny, P., & Kodali, R. (2013). *Big Data Analytics Using Splunk.* New York: Springer Science+Business Media.

Weiterführende Quellen

[AWS_2013] Amazon Web Services, Inc. . (2013). *Amazon Dynamo DB*. Abgerufen am 17. Dezemeber 2013 von http://aws.amazon.com/de/dynamodb/

[AN_2012] Arndt, N. (15. April 2012). *Key-Value-Stores Am Beispiel von Scalaris*. Abgerufen am 17. Dezember 2013 von http://dbs.uni-leipzig.de/file/seminar_1112_arndt_ausarbeitung.pdf

[BT_2011-2013] Basho Technologies, Inc. (2011-2013). *Eventual Consistency*. Abgerufen am 17. Dezember 2013 von http://docs.basho.com/riak/latest/theory/concepts/Eventual-Consistency/

[CIT_2013] Citrusbyte. (2013). *Redis*. Abgerufen am 17. Dezember 2013 von http://redis.io/

[FDB_2013] Foundation DB. (2013). *The CAP Theorem*. Abgerufen am 2013. Dezember 17 von https://foundationdb.com/white-papers/the-cap-theorem

[GH_2013] GitHub. (2013). *Project Voldemort*. Abgerufen am 17. Dezember 2013 von A distributed Database: http://www.project-voldemort.com/voldemort/

[RD_2013] Rosenthal, D. (2. November 2013). *GIGAOM*. Abgerufen am 17. Dezember 2013 von Next gen NoSQL: The demise of eventual consistency?: http://gigaom.com/2013/11/02/next-gen-nosql-the-demise-of-eventual-consistency/

[SIT_2012-2014] Solid IT. (2012-2014). *DB Engines* . Abgerufen am 17. Dezember 2013 von CAP Theorem: http://db-engines.com/de/article/CAP+Theorem

[VW_2007] Voegels, W. (19. Dezember 2007). *All things Distributed*. Abgerufen am 17. Dezember 2013 von Eventually Consistent: http://www.allthingsdistributed.com/2007/12/eventually_consistent.html

Abbildungsverzeichnis

Tabellenverzeichnis

Erklärung

„Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben.“ Alle stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen teilen noch keiner anderen Prüfungsbehörde vorgelegen.

…… ……..

(Ort/Datum) (Unterschrift)