University of Applied Sciences
HOCHSCHULE EMDEN·LEER

Big Data Processing mit Apache Spark

vorgelegt von

Sascha P. Lorenz

Matrikel-Nr.: 501 63 21

dem Fachbereich Technik
der Hochschule Emden-Leer
und der Beuth Hochschule für Technik Berlin
vorgelegte Masterarbeit
zur Erlangung des akademischen Grades
Master of Science (M.Sc.)
im Studiengang
Medieninformatik-Online (Master)

Tag der Abgabe 07. März 2015

1. Betreuer	Herr Prof. Dr. Edlich	Beuth Hochschule für Technik
2. Betreuer	Herr Prof. Dr. Schiemann-Lillie	Hochschule Emden-Leer

Kurzfassung

Gegenstand dieser Arbeit sind die Grundlagen der Verarbeitung und Analyse großer Datenmengen (Big Data) am konkreten Beispiel von Apache Spark. Zunächst sollen verschiedene Ansätze mit Ihren Funktionsweisen sowie den Vor- und Nachteilen diskutiert werdem. Hier werden zuerst allgemeine Grundlagen zu Big Data erarbeitet. Was ist Big Data, was unterscheidet die Verarbeitung von strukturierten und unstrukturierten Daten, Relationale Datenbanken vs. noSQL, wie müssen die Quelldaten für die jeweiligen Verarbeitungen beschaffen sein, welche besonderen Herausforderungen stellen gestreamte Daten an die Verarbeitung. Besonders wird hier auf Hadoop und den Map/Reduce-Algorithmus eingegangen, um das bisher etablierte Vorgehen zu beschreiben und ein grundsätzliches Verständnis für die Domäne "Big Data Processing" zu schaffen. In diesem Kontext wird das gesamte Ökosystem rund um Hadoop vorgestellt.

Nachdem eine Einführung in das Thema "Big Data Processing" erfolgt ist und ein entsprechend quantitativ und qualitativ brauchbarer Datensatz zur Verfügung steht, werden die Next-Generation Data-Processing Technologien betrachtet. Kernthema ist hier Apache Spark und der gesamte BDAS (Berkeley Data Analytics Stack), der von den den AMP-Labs innerhalb von Apache-Projekten um Spark herum aufgebaut wurde. Zu praktisch jeder "offiziellen"BDAS-Implementierung existieren noch Alternativen. Besonders Apache flink wird hier als Alternative näher untersucht. Auch Applikationen, die auf dem eigentlichen Stack aufsetzen, werden näher betrachtet und entsprechenden Praxistests unterzogen (beispielsweise H2O für statistische Analysen).

Danach wird die API von Spark und deren Möglichkeiten mit Scala, Java und Clojure näher betrachtet und durch jeweils eigene Implementierungen untersucht.

Die Arbeit schließt mit durch verschiedene Versuchsreihen fundierte Empfehlungen für die unterschiedlichen Anforderungen im Bereich des Big-Data-Processing.

Abstract

Inhaltsverzeichnis

Kapitel 1

Einführung

"Big Data" ist insbesondere in den letzten Jahren immer stärker in den verschiedensten Zusammenhängen in den allgemeinen Sprachgebrauch vorgedrungen und ist hier einem ständigen Bedeutungswandel ausgesetzt. Besonders in letzter Zeit wird dieses Thema auch verstärkt kontrovers diskutiert.

Im ersten Kapitel soll der Begriff „Big Data" jenseits von Management-Hype und Skepsis rational definiert werden. Des Weiteren werden einige grundlegende Konzepte des Umgangs mit sehr großen und unstrukturierten Datensätzen diskutiert und im Speziellen die Motivation hinter den Apache Frameworks Hadoop und Spark vorgestellt.

Das zweite Kapitel beschäftigt sich mit dem Berkeley Data Analytics Stack (BDAS), mit dem von der UCLA Berkeley rund um Hadoop ein leistungsfähiger Infrastruktur-Stack für die Einsatzbereiche von Big Data Analytics geschaffen wurde.

Innerhalb vom BDAS etabliert sich langsam auch eine schnellere und flexiblere Alternative zu Hadoop: Apache Spark. Im dritten Kapitel wird diese neue Kerntechnologie vorgestellt, die zugleich auch den Hauptteil dieses Wissenschaftlichen Projektes darstellt.

Im vierten Teil dieser Ausarbeitung wird Spark in der praktischen Anwendung gezeigt inklusive Installation und ersten kleineren Beispielen sowohl direkt in Spark, als auch aus darüber liegenden Schichten aus dem Stack.

1.1 Was versteht man unter "Big Data"?

Der Begriff "Big Data" wurde vermutlich zum ersten Mal Ende des 20. Jahrhunderts von John R. Marshey, damals Chefwissenschaftler bei Silicon Graphics, im Rahmen einer Usenix-Konferenz öffentlich erwähnt. Mittlerweile ziert dieser Begriff gefühlt jedes zweite Cover von IT-Zeitschriften mit Business-Fokus und auch Manager und "Sales-Professionals" werten Ihre Produktpräsentationen gerne mit diesem Buzzword auf. Aber dieser Begriff ist nicht nur positiv assoziiert. Besonders seit Bekanntwerden der Tätigkeiten des Amerikanischen Auslandsgeheimdienstes weckt die Vorstellung des „Datensammelns" in großen Dimensionen auch Misstrauen.

Im Rahmen dieser Ausarbeitung soll jedoch ausschließlich die technische Betrachtung und die exemplarische Darstellung von möglichen Anwendungsgebieten diskutiert werden.

Wie lässt sich der Begriff „Big Data“ abgrenzen? Es existiert keine abschließend eindeutige Definition, jedoch gibt es einige Attribute, die sich in einem Großteil der Fachliteratur etabliert haben. Der Wikipedia-Artikel zum Thema [WP14] fasst dies folgendermaßen zusammen:

„[. . .]bezeichnet Datenmengen, die zu groß, oder zu komplex sind, oder sich zu schnell ändern, um sie mit händischen und klassischen Methoden der Datenverarbeitung auszuwerten.“

Neben der reinen Menge spielt also offensichtlich auch die mangelnde oder fehlende Strukturierung der Daten eine nicht unerhebliche Rolle. Dies können beispielsweise Daten aus Social-Media-Quellen sein, die aus allen möglichen verschiedenen Einzeldaten bestehen, Daten von Sensoren, die permanent überwacht werden müssen, oder Datenströme (Video, Audio, Bilder, Text), die nach einheitlichen Kriterien gefiltert werden sollen, um hier nur einige Beispiele zu nennen. Auch die temporäre Komponente ist ein Einsatzgebiet für „Big Data“, und auch hier ist wieder das Beispiel der Datenströme heranzuziehen.

Bei der Definition von Big Data werden laut des BITKOM-Ratgebers zum Thema „Big Data“ [BK14] auch immer wieder die „Three Vs“ angeführt. Dies sind Volume, also die Datenmenge, Variety, die Datenvielfalt und Velocity, die Geschwindigkeit der Auswertung.

Die sinnvolle Analyse dieser Daten kann Unternehmen oder anderen Organisationen wichtige Informationen z.B. über Marktentwicklungen, bestimmte Kundenbedürfnisse, Epedemie-Ausbreitungen oder andere wichtige Sachverhalte liefern. Diese Analyse inklusive der dazu verwendeten Werkzeuge wird allgemein „Big Data Analytics“ genannt.

1.2 Ansätze für “Big Data Analytics“

“Big Data Analytics“ umfasst Methoden und Werkzeuge zur automatisierten oder interaktiven Erkennung und daraufhin auch Verwendung von bestimmten Mustern und Assoziationen. Dies sind unter anderem:

- Prediction-Models zur Vorhersage bestimmter Sachverhalte
- statistische Verfahren, wie beispielsweise Logistic Regression oder k-means-Algorithmen
- Optimierungs- und Filteralgorithmen
- Werkzeuge zum Datamining
- Textanalyse
- Bild- und Tonanalyse
- Datenstromanalysen

Nach dem BITKOM-Leitfaden [BK14] besteht die Taxonomie der Big-Data-Technologien grundsätzlich aus vier Schichten:

- Daten-Haltung
- Daten-Zugriff

- Analytische Verarbeitung
- Visualisierung

Diese werden durch Daten-Integration und Daten-Governance, sowie Daten-Sicherheit flankiert, um den Weg von Rohdaten bis zu nutzbaren Erkenntnissen in existierende Standards einzubetten.

Zahlreiche Hersteller herkömmlicher relationaler Datenbanksysteme versuchen derzeit, ihre bestehenden Lösungen mit dem Label „Big Data“ zu versehen und diese so weiterhin in diesen sich verändernden Marktsegmenten zu positionieren. Wenn „Big Data“ jedoch jenseits der Datengröße definiert wird und auch unstrukturierte und temporäre Daten-Stacks oder –ströme zu verarbeiten oder zu analysieren sind, stoßen RDBMS sehr schnell an ihre Grenzen. Doch auch was die Skalierbarkeit angeht, sind relationale Datenbanken meist nicht hinreichend flexibel.

Für die Anforderungen an dedizierte Aufgaben im Bereich Big-Data-Analytics sind seit einigen Jahren einige Frameworks auf dem Markt, die in allen drei oben genannten Aspekten besser geeignet sind, als RDBMS. Der Ansatz ist hier primär, die Verarbeitung zu dezentralisieren, also auf unabhängige Knoten in einem Rechner-Cluster zu verteilen und nur Referenzen auf die Clusterknoten zentral zu verwalten.

Es existieren mittlerweile Lösungen am Markt, die speziell diese Aufgaben für derartige Aufgaben entwickelt wurden. Hier wären unter anderem Hadoop, Spark, HPCC, GPMR, Mincmeat, Sphere, Bashreduce und R3 zu nennen. Bis auf HPCC setzen alle eben genannten Implementierungen generell oder in Teilen auf das Programmiermodell MapReduce.

Der zweifellose De-facto-Standard in diesen Bereichen ist bereits seit einiger Zeit das Open-Source-Framework Apache Hadoop. Auf Hadoop basierend existieren etliche Derivate. Unter anderem sind hier Cloudera, Amazon Elastic MapReduce, Apache BigTop, Datameer, Apache Mahout, MapR und IBM PureData System zu nennen.

1.3 Motivation für Apache Hadoop/Spark

Anfang des 21. Jahrhunderts wurde das Bedürfnis für Möglichkeiten, sehr große Datenmengen effizient verarbeiten zu können, stetig größer. Nicht zuletzt durch die zu dieser Zeit exponentiell steigende Menge von Inhalten im World Wide Web und deren Indexierung durch Suchmaschinen wie Google. Davon motiviert wurde 2002 das Projekt „Nutch“ mit dem Ziel gestartet, ein geeignetes Such- und Crawlersystem frei verfügbar zu machen. Die ersten Versuche skalierten sehr schlecht, bis Google 2003 die Funktionsweise ihres verteilten Dateisystem GFS (Google File System) veröffentlichte. Somit konnten die sehr großen Dateien, die durch die Indexierung entstanden, effizient auf verschiedene Knoten verteilt gespeichert werden und die Verwaltung dieser Knoten und Dateien aus dem eigentlichen Indexierungs- und Suchprozess ausgelagert werden.

Im Jahre 2004 publizierte Google den MapReduce-Algorithmus, der unter anderem die Indexierungs- und Analysefunktionen parallelisieren, delegieren und sinnvoll bündeln kann. In Nutch wurden daraufhin sämtliche wichtige Algorithmen auf MapReduce umgestellt, nachdem zuvor auch GFS unter dem Namen NDFS (Nutch Distributed File System) integriert wurde. Die möglichen Anwendungsgebiete von Nutch waren damit auch weit über das reine Suchen und Indexieren von

Webseiten hinaus gewachsen. 2006 wurde aus Nutch ein Unterprojekt mit dem Namen Hadoop ausgegliedert, das im Jahre 2008 zum Apache Top-Level-Project ernannt wurde. Zu dieser Zeit nutzten bereits Firmen wie Yahoo!, Facebook oder die New York Times Hadoop. Ein exemplarischer Anwendungsfall bei der NY Times war, mit Hilfe der Hadoop-basierten EC2-Cloud von Amazon ca. vier Terabyte gescannter Archivdateien in PDF-Dateien umzuwandeln und dies in weniger als 24 Stunden auf 100 Knoten. Auch beim Sortieren von sehr großen Datenmengen stellten Hadoop-basierte Systeme nach und nach sämtliche Rekorde ein [TW09].

Hadoop und Hadoop-basierte System gelten mittlerweile als Industriestandard für Big-Data-Analytics-Anwendungen. Jedoch ist Hadoop nicht für alle Anwendungsgebiete gleichermaßen geeignet. Aufgrund der Charakterisierung der Paradigmen für Big Data Analytics im Paper „Frontiers in Massive Data Analysis“ der National Academic Press [NRC13], lassen sich die Einsatzgebiete und Schwächen für Hadoop ermitteln [VA14].

So lassen sich mit Hadoop einfachere statistische Aufgabenstellungen sehr gut umsetzen. Dazu gehören Mittelwert, Median, Varianz und allgemein abzählende sowie ordnende Statistikaufgaben. Dies sind in der Regel Anwendungen mit einer Laufzeitkomplexität von O(n) für n Betrachtungswerte. Sie sind meist auch sehr gut parallelisierbar und somit sehr gut für Hadoop geeignet.

Für linear-algebraische Berechnungen (lineare Regression, Eigenwertproblem, Hauptkomponentenanalyse), generalisierte n-Körper-Probleme (mit einer Komplexität von $O(n^2)$ oder $O(n^3)$), Graphentheorie, Optimierungsprobleme (Verlust-, Kosten- oder Energiefunktionen, sowie Integrations- und Ausrichtungsfunktionen ist Hadoop nur in jeweils einfacher Problemausprägung einsetzbar. Auch für Interaktive Abfragen ist Hadoop nur bedingt geeignet, da es ursprünglich für die Batch-Verarbeitung entwickelt wurde.

Aus diesem Grund wurde am AMPLab der University of California in Berkeley nach Alternativen geforscht, die auch für komplexe linear-algebraische Probleme, generalisierte n-Körper-Probleme und diverse Optimierungsprobleme geeignet sind. Das Ergebnis ist Spark, mittlerweile Apache Top-Level-Projekt und dazu geeignet, die Nachfolge von Hadoop als Big-Data-Analytics-Framework anzutreten.

Kapitel 2

Verzeichnisse

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Anhang A

Zusätze

A.1 Quelltext