



Vergleich dreier Implementationsvarianten für eine Analyse von Satellitenbildern

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Arts (B. A.)

HUMBOLDT-UNIVERSITÄT ZU BERLIN
MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT II
INSTITUT FÜR INFORMATIK

eingereicht von: Robin Ellerkmann
geboren am: 25.04.1992
in: Berlin

Gutachter: Prof. Johann-Christoph Freytag, Ph.D.
Dipl.-Inf. Mathias Peters

eingereicht am:

verteidigt am:

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	2
2.1	Grundlagen zur Satellitenbildanalyse	2
2.1.1	Geographische Definitionen zur Satellitenbildanalyse	2
2.1.2	Beschreibung des Algorithmus	2
2.2	Maschinelle Grundlagen	2
2.2.1	Parallelisierte Systeme und Big Data	2
2.2.2	Apache Flink	3
2.2.3	Python	4
3	Implementation des Algorithmus	5
3.1	Umsetzung des Algorithmus mit Apache Flink	5
3.1.1	Nutzung der Java-Programmierschnittstelle	5
3.1.2	Nutzung der Python-Programmierschnittstelle	5
3.2	Umsetzung des Algorithmus in Python	5
4	Tests	7
4.1	Versuchsbeschreibung	7
4.2	Auswertung	7
5	Fazit	8

Kapitel 1

Einleitung

Seit einigen Jahren ist ein massiver [Anstieg](#) an [Datenaufkommen](#) zu beobachten [EMC14]. Diese Entwicklung erfordert neue Technologien und Prozesse, zum Beispiel bei der Speicherung und der Verarbeitung der Daten. Denn traditionelle Datenbanksysteme können große Datenmengen nicht immer in akzeptabler Form und Verarbeitungszeit verarbeiten [Jac09]. Ein zum Zweck der Verarbeitung großer Datenmengen entwickeltes Programmiermodell ist das Map-Reduce Paradigma, das 2004 erstmals publiziert wurde [DG08]. Dieses Paradigma sieht eine massiv parallelisierte Verarbeitung von Daten vor und wird von Datenverarbeitungssystemen wie Hadoop [Foub] und Flink [Foua] implementiert.

Im Rahmen dieser Bachelorarbeit sollen ein traditioneller Ansatz und ein massiv parallelisierbarer Ansatz bei der Verarbeitung von großen Datenmengen untersucht werden. Der Vergleich beider Ansätze wird am Beispiel eines Algorithmus zur Approximierung einer Pixelzeitreihe durchgeführt. Dieser wird im Rahmen des Projekts GeoMultiSens [GP] zur Analyse der Veränderung der Flora in einer geographischen Region genutzt. An die Analyse anschließend werden mithilfe des Algorithmus auf Basis der approximierten Werte Prognosen zur weiteren Entwicklung der Flora der untersuchten Region gestellt.

Es werden drei unterschiedliche Implementierungen des Algorithmus untersucht, die sich hinsichtlich der eingesetzten Technologien und Programmiersprachen unterscheiden. Die Methodik, die der Algorithmus implementiert, ist bei allen untersuchten Varianten identisch. Als Basis wird die bereits implementierte und in der Praxis genutzte Python-Implementation genutzt. Die zweite und dritte Variante werden in Flink implementiert. Diese beiden Varianten unterscheiden sich bezüglich der genutzten Programmiersprache. [Zur Implementierung von Variante zwei wird Flinks Java-Schnittstelle genutzt, zur Umsetzung von Variante drei die Python-Schnittstelle.](#) Schließlich werden alle drei Varianten unter identischen Bedingungen getestet. Dies bedeutet, dass sowohl die Testumgebung als auch die Testdaten identisch sein sollen. Ausgehend von den Tests und den ermittelten Ergebnissen wird eine Bewertung der drei Implementierungsvarianten des Algorithmus vorgenommen werden.

Kapitel 2

Grundlagen

2.1 Grundlagen der Satellitenbildanalyse

2.1.1 Geographische Definitionen zur Satellitenbildanalyse

Einführung von geo. Dingen (Koordinaten(Definition, numerische Darstellung, Umgang mit Koordinaten), Aufbereitung von Bildern (Transformieren von Bildern zur Entzerrung.), Fernerkundung etc.)

2.1.2 Beschreibung des Algorithmus zur Analyse von Satellitenbildern

Beschreibung der Vorgehensweise bei der Analyse (Zhu, SVR), Ziel der Analyse, Entwicklungsgeschichte der Analysetechnik

2.2 Maschinelle Grundlagen

2.2.1 Parallelisierte Systeme und Big Data

Um die seit mehreren Jahren massiv ansteigenden Datenmengen [EMC14] zu verarbeiten wird zunehmend eine verteilte Verarbeitung dieser Daten populär. Dazu werden mehrere Maschinen zu einem Netzwerk, einem sogenannten Cluster, zusammengeschlossen. Diese Computer wären als einzelne Maschine nicht in der Lage ein großes beziehungsweise komplexes Problem in akzeptabler Zeit zu lösen. Die Leistungsfähigkeit des Netzwerks wird jedoch nicht über die Leistung einer einzelnen Maschine sondern primär über die Menge der zusammengeschlossenen Computer gesteuert. Dies hat mehrere Vorteile gegenüber der Verarbeitung mithilfe einzelner, besonders leistungsstarker Maschinen. Die wichtigsten Vorteile parallelisierter Systeme sind ihre Skalierbarkeit sowie die Fehlertoleranz. Falls mehr Rechenleistung benötigt wird oder wenn Teile des Netzwerks nicht funktionsfähig sind lassen sich neue Maschinen kurzfristig, meist auch im laufenden Betrieb, in das bestehende Netzwerk integrieren. Bei einzelnen, sehr leistungsstarken Computern gestaltet sich beides aufgrund der abgeschlossenen Beschaffenheit der Maschine schwierig. [Quelle]

Eine **Big-Data Anwendung** zeichnet sich durch drei Eigenschaften aus. Diese drei Charakteristika sind die Größe (engl. Volume), die Komplexität (eng. Variety) und die echtzeitnahe Verfügbarkeit sowie schnelle Verarbeitung (engl. Velocity) der Daten [Lan01]. Ein weiteres Merkmal ist die nicht garantierte Zuverlässigkeit und Einheitlichkeit der Daten (engl. veracity) [ZdP⁺12]. In den letzten Jahren hat sich das global produzierte Datenaufkommen massiv gesteigert. Insbesondere die zunehmende Zahl der Internetnutzer sowie die Verbreitung von Smartphones trägt zu dieser Entwicklung bei. Ebenso trägt die zunehmende Digitalisierung der Industrie sowie die zunehmende Verbreitung von Sensoren jeglicher Art zu diesem Anstieg bei. Aber auch in nicht kommerziellen Bereichen wächst die Datenmenge. Die Satellitenbilder der aktuellen Generation des Landsat-Satellitensystems produzieren Aufnahmen, die dreimal soviel Speicherplatz benötigen wie die der vorigen Generation. Projekte wie das Sloan Digital Sky Survey produzieren täglich etwa 200 Gigabyte. Insgesamt werden die Datenmengen weiter massiv zunehmen. Für das Jahr 2020 wird ein weltweites Datenaufkommen von 44 Zettabyte prognostiziert [EMC14]. Aus dieser steigenden Datenmenge ergeben sich auch Folgen für Daten verarbeitende Dienste. Es müssen sehr viel mehr Daten auf einmal verarbeitet werden. Darüber hinaus sind die zu verarbeitenden Daten zunehmend vielfältiger und unstrukturierter. Die verarbeitenden Algorithmen und die Speicherstrukturen müssen also hinreichend auf unvollständige beziehungsweise fehlerhafte Datensätze reagieren können und diese trotzdem bestmöglich verarbeiten. [Quellen] [Beschreibung für velocity einfügen]. Wenn eine Anwendung eine Datenmenge verarbeitet, die mindestens einige der vier Kriterien nach [Lan01] erfüllt, gilt diese Anwendung als Bi-Data Anwendung.

Eigenschaften von: Big Data, DBMS, Grundlagen für Flink, Erwähnung MapReduce Prinzip

2.2.2 Apache Flink

Eigenschaften + Operatoren in Flink

Apache Flink ist ein System/Framework, das auf eine massiv parallelisierte Verarbeitung **Vorher einführen, 2.2.1** von großen Datenmengen spezialisiert ist. Es ging **2014** [Quelle] aus Stratosphere hervor, das seit 2010 [Quelle] kooperativ von Forschern verschiedener Universitäten entwickelt wurde [ABE⁺14]. Seit Januar 2015 ist Flink ein Top-Level Projekt der Apache Software Foundation [Fou15].

Die Hauptkomponenten des Systems sind die Flink-Laufzeitumgebung und der Flink-Optimierer. Der Flink-Optimierer erhält einen azyklischen Graphen von Flink-Operatoren als Eingabe. Dieser wird mithilfe von Techniken der traditionellen Optimierung von relationalen Anfragen optimiert. [Weitere Details aus Stratosphere Paper?, unter welchen Gesichtspunkten wird DAG optimiert?]. Der optimierte Datenflussgraph, auch Jobgraph genannt, besteht aus mehreren, teilweise unabhängig voneinander zu verarbeitenden Arbeitsschritten. Diese können teilweise parallel bearbeitet werden [MapReduce erwähnen?]. Dieser optimierte Datenflussgraph wird an die Flink-Laufzeitumgebung weitergegeben.

Flink erweitert das Map-Reduce Paradigma um weitere Operatoren. [Operatoren beschreiben]

2.2.3 Python

Python ist eine quelloffene und universell einsetzbare Programmiersprache, die seit 1989 existiert und fortwährend weiter entwickelt wird. Prägende Eigenschaften der Sprache sind unter anderem eine dynamische Typisierung von Variablen, eine simpel gehaltene Syntax und die Erweiterbarkeit durch Module und Bibliotheken. Es ist auch möglich Python-Code durch C- beziehungsweise C++-Bibliotheken zu erweitern [Mar06]. Dies ermöglicht eine verkürzte Ausführungszeit eines Programms, insbesondere bei rechenintensiven Programmabschnitten. Ein Schwachpunkt von Python im Bezug auf die schnelle Verarbeitung großer Datenmengen ist die nicht auf automatisierte Parallelisierung ausgelegte Architektur. Daraus resultiert eine unzureichende Skalierbarkeit, sobald Daten, deren Größe die Arbeitsspeichergröße der ausführenden Maschine übersteigt, verarbeitet werden müssen. (Auf weiter oben genannten Punkt der Großen Datenmengen eingehen). Bez. der Eignung zur Lösung solcher Probleme.

Kapitel 3

Implementation des Algorithmus

3.1 Umsetzung des Algorithmus mit Apache Flink

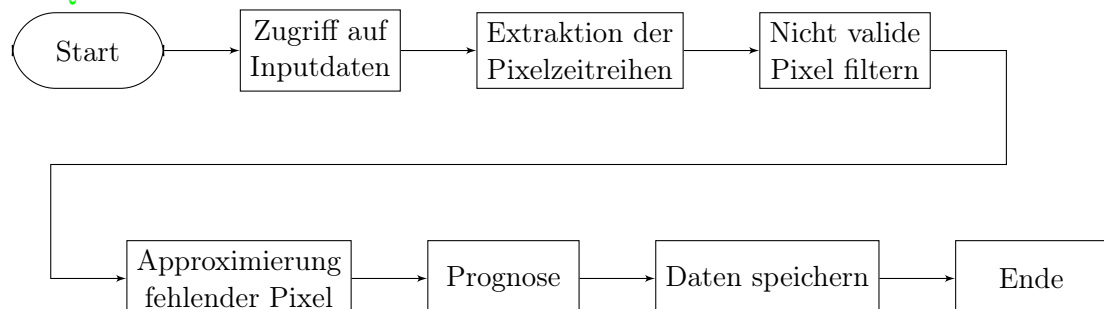
3.1.1 Nutzung der Java-Programmierschnittstelle

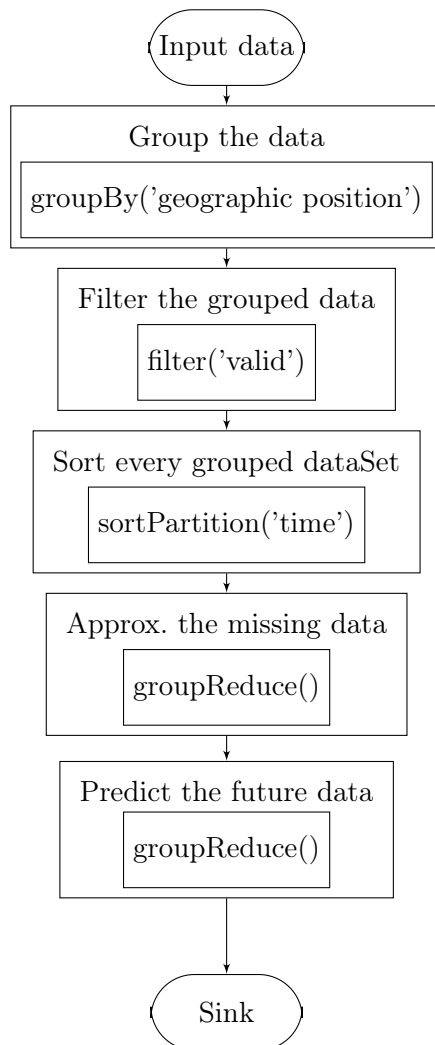
3.1.2 Nutzung der Python-Programmierschnittstelle

3.2 Umsetzung des Algorithmus in Python

Die Analyse von Satellitenbildern erfordert die Verarbeitung großer Mengen komplexer Rohdaten, **die nahezu in Echtzeit verfügbar sind**. Aufgrund dieser Charakteristika handelt es sich bei dieser Analyse um ein **Big-Data Problem**. Denn alle vier Kriterien, die ein solches charakterisieren sind erfüllt.

Bei der Analyse von Satellitenbildern sind die Merkmale Datengröße und Datenkomplexität sowie die schnelle Verarbeitung der Daten von Bedeutung. Abhängig von der Anzahl der genutzten Bilder sind die zu verarbeitenden Datenmengen sehr groß. Ein Bild besitzt im Regelfall abhängig vom Satellitenmodell, das die Aufnahme gemacht hat, eine Größe von 750 Megabyte bis zu 1,5 Gigabyte. Um eine Entwicklung zu untersuchen werden jedoch viele dieser Bilder in die Untersuchung mit einbezogen, so dass die zu verarbeitende Datenmenge kontinuierlich ansteigt. Dieser kontinuierliche Anstieg entsteht dadurch, dass aktuell mehrere Satelliten mit der Fernerkundung der Erde fortfahren und so in kurzen Intervallen neue Bilder zur Verfügung stehen, die im Rahmen der Analyse verwendet werden sollen. **Quelle**.





Kapitel 4

Tests

4.1 Versuchsbeschreibung

Beschreibung + Begründung für meine Versuchsbedingungen

4.2 Auswertung

Beschreibung und Bewertung der Ergebnisse meiner Untersuchungen

Kapitel 5

Fazit

Fazit und Ausblick z.b. Vergleich mit anderen Untersuchungen

Literaturverzeichnis

- [ABE⁺14] Alexander Alexandrov, Rico Bergmann, Stephan Ewen, Johann-Christoph Freytag, Fabian Hueske, Arvid Heise, Odej Kao, Marcus Leich, Ulf Leser, Volker Markl, and et al. The stratosphere platform for big data analytics. *The VLDB Journal*, 23(6):939,964, May 2014.
- [DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce. *Communications of the ACM*, 51(1):107, Jan 2008.
- [EMC14] EMC². The digital universe of opportunities. Technical report, EMC², 2014.
- [Foua] Apache Software Foundation. Flink website. <https://flink.apache.org/>.
- [Foub] Apache Software Foundation. Hadoop website. <https://hadoop.apache.org/>.
- [Fou15] Apache Software Foundation. The apache software foundation announces apacheTM flinkTM as a top-level project. https://blogs.apache.org/foundation/entry/the_apache_software_foundation_announces69, January 2015.
- [GP] GfZ-Potsdam. Geomultisens website. <http://www.geomultisens.gfz-potsdam.de/>.
- [Jac09] Adam Jacobs. The pathologies of big data. *Communications of the ACM*, 52(8):36, August 2009.
- [Lan01] Doug Laney. 3d data management: Controlling data volume, velocity and variety. *Application Delivery Strategies published by META Group Inc.*, Feb 2001.
- [Mar06] Alex Martelli. *Python in a Nutshell. A Desktop Quick Reference*. O'Reilly, second edition edition, 2006.
- [ZdP⁺12] Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, and David Corrigan. *Harness the Power of Big Data The IBM Big Data Platform*. McGraw-Hill Osborne Media, 2012.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Weiterhin erkläre ich, eine ...arbeit in diesem Studienggebiet erstmalig einzureichen.

Berlin, den 29. Juli 2015

.....

Statement of authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such. Neither this nor a similar work has been presented to an examination committee.

Berlin, 29th July 2015

.....