



# 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

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Grundlagen der Satellitenbildanalyse . . . . .	3
2.1.1	Fernerkundung mithilfe des Landsat-Satellitensystems . . . . .	3
2.1.2	Aufbereitung und Analyse von Satellitenbildern . . . . .	4
2.2	Parallele Datenverarbeitungssysteme . . . . .	5
2.2.1	Bedeutung und Eigenschaften von Big Data . . . . .	5
2.2.2	Verarbeitung großer Datenmengen - Konzepte und Systeme . . . . .	5
2.2.3	Apache Flink . . . . .	5
2.2.4	Python . . . . .	5
<b>3</b>	<b>Algorithmus zur Analyse von Pixelzeitreihen</b>	<b>7</b>
3.1	Beschreibung des Algorithmus . . . . .	7
3.2	Umsetzung des Algorithmus mit Apache Flink . . . . .	7
3.2.1	Nutzung der Java-Programmierschnittstelle . . . . .	7
3.2.2	Nutzung der Python-Programmierschnittstelle . . . . .	7
3.3	Umsetzung des Algorithmus in Python . . . . .	7
<b>4</b>	<b>Evaluierung</b>	<b>9</b>
4.1	Versuchsbeschreibung . . . . .	9
4.2	Auswertung . . . . .	9
<b>5</b>	<b>Fazit</b>	<b>10</b>

# Kapitel 1

## Einleitung

In den vergangenen Jahren war ein massiver Zunahme des generierten Datenaufkommens zu beobachten [EMC14]. Viele Projekte, Unternehmen und Institutionen haben Zugriff auf eine gewaltige Menge an Daten. Diese wächst immer schneller an. 2004 analysierte Google circa 100 Terabyte pro Tag [DG04]. Bis zum Jahr 2008 war die täglich zu analysierende Datenmenge bereits auf 20 Petabyte angewachsen [DG08]. Das Sloan Digital Sky Survey, das ein Viertel des Himmels astronomisch erkundet, hat seit 1998 insgesamt 116 Terabyte an astronomischen Daten gesammelt [YAJEA<sup>+</sup>00, AAP<sup>+</sup>15]. Jede Nacht kommen circa 250 Gigabyte neu hinzu [Quelle: Herr Prof. Freytags VL, dort angegebene Quelle ist offline. Wie angeben?](#). Ein weiteres Beispiel ist das 1000 Genomes Project [Bak10], das zwischen 2008 und 2013 insgesamt 464 Terabyte Daten zum menschlichen Genom sammelte. Insgesamt werden die Datenmengen weiter stark zunehmen, für das Jahr 2020 wird eine weltweites Datenaufkommen von 44 Zettabyte prognostiziert [EMC14]. Diese Entwicklung offenbart diverse neue Herausforderungen bei der Speicherung, Verarbeitung und Analyse von Daten. Dabei spielt die möglichst schnelle Verarbeitung von stetig generierten Daten eine große Rolle. Diese muss im Gegensatz zur Verarbeitung bereits gespeicherter Daten abhängig vom aktuellen Datenaufkommen skalieren. Aktuelle Datenverarbeitungssysteme wie Apache Hadoop [Foud] und Apache Flink [Fouc] bieten diese Möglichkeit der Datenflussanalyse und ermöglichen eine flexible Analyse der Daten. Kern dieser Systeme ist eine Implementierung des Map-Reduce Paradigmas [DG08] sowie die Nutzung von *User Defined Functions*. Diese ermöglichen eine parallele Abarbeitung von Arbeitsschritten in einem direkten, azyklischen Graphen. Der DAG wird zuvor aus dem vom User bereitgestellten Quellcode erzeugt. Die durch diese Architektur erreichbare, massiv parallelisierbare Ausführung der Datenanalyse ermöglicht die Nutzung von Clustern. Somit wird eine skalierbare Infrastruktur genutzt, die wiederum eine skalierte Nutzung der Datenverarbeitungssysteme ermöglicht. Diese Systeme können auch die weiterhin wichtige Nutzung und Analyse von bereits konsistent gespeicherten Daten unter Nutzung des parallelen Verarbeitungsansatzes durchführen. Dies ist insbesondere deshalb notwendig, da traditionelle Datenbanksysteme große Datenmengen nicht immer in akzeptabler Form und Verarbeitungszeit verarbeiten können [Jac09]. Das Hauptproblem bei der Verarbeitung von großen Datenmengen auf einzelnen Maschinen entsteht, wenn die zu verarbeitende Datenmenge die Hauptspeichergröße übersteigt. In diesem Fall müssen die nicht in den Hauptspeicher speicherbaren Daten zur Verarbeitungszeit nachgeladen werden, was die Verarbeitungszeit aufgrund der unterschiedlichen Beschaffenheit der verschiedenen Speicherebenen extrem verlängert. Um diese Speicherklappe zu umgehen, werden zunehmend parallelisierbare Ansätze der Datenverarbeitung verfolgt. Im Rahmen dieser Bachelorarbeit sollen demzufolge ein traditioneller und ein massiv parallelisierbarer Ansatz bei der Verarbeitung von großen Datenmengen untersucht werden. So soll eine Abschätzung der Leistungsfähigkeit, Vorteile und Nachteile beider Ansätze ermittelt werden. Der Vergleich beider Ansätze wird am Beispiel ei-

nes Algorithmus zur Approximierung von Pixelzeitreihen durchgeführt. Dieser wird im Rahmen des Projekts GeoMultiSens[GP] zur Analyse der Veränderung der Flora in einer geographischen Region genutzt. Dabei werden durch Landsat-Satelliten Satellitenaufnahmen bereitgestellt, die nach der Aufbereitung durch vorgestellte Algorithmen ausschnittsweise untersucht werden. Nach der Analyse werden anschließend mithilfe des Algorithmus auf Basis der approximierten Werte Prognosen zur weiteren Entwicklung der Flora der untersuchten Region gestellt. Dabei werden bei einer Analyse mehrere Szenenausschnitte derselben geographischen Region analysiert. Dabei müssen große Datenmengen verarbeitet werden, so dass sich die Nutzung eines aktuellen Datenverarbeitungssystems anbietet. Bei dieser Bachelorarbeit wird als Vertreter der massiv parallelisierbaren Datenverarbeitungssysteme Apache Flink genutzt.

Es werden drei unterschiedliche Implementierungen des Algorithmus untersucht, die sich hinsichtlich der eingesetzten Technologien und Programmiersprachen unterscheiden. Die zugrunde liegende 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. Sie sollte somit die untere Schranke der Leistungsmessungen darstellen. Die zweite und dritte Variante werden in Flink implementiert. Diese beiden Varianten unterscheiden sich bezüglich der genutzten Programmiersprache. [Zur Implementierung der zweiten Variante wird Java-Schnittstelle von Flink 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. Dabei sollen alle Varianten sowohl auf einer leistungsfähigen Einzelmaschine als auch mit einem Cluster von Maschinen getestet werden. Ausgehend von den Untersuchungen und den ermittelten Ergebnissen soll nachfolgend eine Bewertung der drei Implementierungsvarianten des Algorithmus vorgenommen werden. Dabei sollen insbesondere die Größe der Ausgangsdatenmenge, die genutzte Hardware sowie die Größe der untersuchten Bildausschnitte in Bezug zu den Ergebnissen gesetzt werden.

# Kapitel 2

## Grundlagen

Die Basis für die Satellitenbildanalyse mittels Apache Flink bilden zum einen Konzepte aus der Geographie, zum anderen Strukturen und Vorgehensweisen aus der Informatik. Während die geographische Komponente insbesondere bei der Aufnahme der Satellitenbilder sowie bei der inhaltlichen Konzeption der Analysen sowie der Bereinigung der Daten vertreten ist, ist die Informatik für eine technisch korrekte und effiziente Umsetzung der geographischen Konzepte verantwortlich. Nachfolgend werden zuerst die geographischen Grundlagen der Fernerkundung erläutert. Dies umfasst insbesondere die technische Spezifikation der Aufnahmegeräte der eingesetzten Satelliten sowie wichtige Verfahren zur Datenaufbereitung sowie zur Datenanalyse. Anschließend wird ein Überblick über parallele Datenverarbeitungssysteme gegeben. Dieser umfasst auch die Eigenschaften von Big Data sowie eine konzeptuelle Beschreibung der Datenanalyseplattform Apache Flink. Abschließend wird Apache Flink aus der Entwicklerperspektive betrachtet. Dabei werden insbesondere Eigenschaften der Plattform beschrieben, die bei der Entwicklung von Programmen auf Basis von Flink von Bedeutung sind. Des weiteren wird die Programmiersprache Python betrachtet.

### 2.1 Grundlagen der Satellitenbildanalyse

#### 2.1.1 Fernerkundung mithilfe des Landsat-Satellitensystems

Als Fernerkundung wird „die Gesamtheit der Verfahren zur Gewinnung von Informationen über die Erdoberfläche oder anderer nicht direkt zugänglicher Objekte durch Messung und Interpretation der von ihr ausgehenden (Energie-) Felder“ [Deu12] verstanden. Fernerkundungssatelliten verfügen über verschiedene Aufnahmesysteme, die durch multispektrale Messungen von emittierter elektromagnetischer Strahlung eine berührungsfreie Beobachtung der Erdoberfläche ermöglichen. Bei der multispektralen Messung werden von Sensoren registrierte spektrale Signaturen einzelnen Bereichen des elektromagnetischen Spektrums zugeordnet. Das Resultat sind mehrere spektrumsspezifische, simultan aufgenommene Satellitenbilder, die nur das aufgefangene Licht eines spezifischen Spektralbereichs, auch Spektralband genannt, zeigen. Die Art und Qualität der Aufnahmesensoren ist dabei abhängig vom Typ des Satelliten.

Die Ausgangsdaten für die Untersuchungen in dieser Bachelorarbeit wurden von Satelliten des Landsat-Satellitensystems aufgenommen. Der erste Landsat-Satellit Landsat 1 wurde 1972 gestartet. Seitdem wurden die Sensoren und die Satelliten kontinuierlich weiterentwickelt. Aktuell sind Landsat 7 und, im Rahmen der Landsat Data Continuity Mission, Landsat 8 im Einsatz. Landsat 8 nutzt in der aktuellen Generation zwei verschiedene Instrumente zur Fernerkundung. Den Operational Land Imager (OLI) und die Thermal Infrared Sensors (TIRS).

Der OLI erfasst emittierte elektromagnetische Strahlung im Spektralbereich von 0,433  $\mu\text{m}$  bis 1,390  $\mu\text{m}$  unterteilt in acht Spektralkanäle sowie einen panchromatischen Kanal. Es werden mehr als 7000 Detektoren pro Spektralband genutzt, um eine bessere Bildqualität zu bieten als frühere Systeme [MSWI04]. Neben den klassischen Farbspektren Blau, Grün und Rot nutzt Landsat-8 ein weiteres Band, das speziell für die Fernerkundung von Küsten genutzt wird. Außerdem verfügt Landsat-8 über drei Infrarotbänder, die nahes und mittleres Infrarotlicht registrieren, sowie ein weiteres Infrarotband, das auf die Beobachtung von Cirruswolken spezialisiert ist. Der panchromatische Kanal registriert elektromagnetische Strahlung mit Wellenlängen von 0,500  $\mu\text{m}$  bis 0,680  $\mu\text{m}$ . Dieser Spektralbereich entspricht etwa dem des menschlichen Auges. Aufgrund des, im Vergleich zu den einzelnen Farbfrequenzbändern, breiten abgedeckten Spektralbereichs ist eine höhere Auflösung der Bilder möglich.

Die Thermal Infrared Sensors (TIRS) [Cha11] umfassen zwei Thermalkanäle. Diese erfassen im Gegensatz zu den Multispektralkanälen elektromagnetische Emissionen mit Wellenlängen zwischen 10,30  $\mu\text{m}$  und 12,50  $\mu\text{m}$ , also langwellige Infrarotstrahlung. Dies ist insbesondere für die Beobachtung von Wolken nützlich. Die Kantenlänge der einzelnen Pixel beträgt 100 Meter. Diese kann nachträglich auf 30 Meter angeglichen werden, um eine bessere Kompatibilität mit den Aufnahmen der Multispektralbänder zu gewährleisten.

Landsat 8 sendet pro Tag 400 Aufnahmen der Erdoberfläche, auch Szenen genannt, an die Bodenstation. Eine Aufnahme zeigt dabei eine geographische Region der Erde mit einer Ost-West-Ausdehnung von 185 Kilometer. Dies entspricht 100 nautischen Meilen. Die Nord-Süd-Ausdehnung einer Szene beträgt circa 174 Kilometer

Durchschnittlich wird jede Region der Erde alle 16 (?) Tage überflogen [IDB12].

Die von Landsat-Satelliten aufgezeichneten und übermittelten Bilder müssen jedoch vor der Durchführung von Analysen aufbereitet werden.

### 2.1.2 Aufbereitung und Analyse von Satellitenbildern

Die durch die Landsat-Satelliten aufgezeichneten und an die Bodenstationen übermittelten Szenen müssen vor ihrer Nutzung aufbereitet werden. Dadurch wird im Allgemeinen die Bildqualität verbessert, da externe Störfaktoren und eventuelle interne Fehlfunktionen ausgeglichen werden können. Es wird zwischen radiometrischen und die geometrischen Aufbereitungen unterschieden. Bei der radiometrischen Aufarbeitung werden digitale Werte wie zum Beispiel die Helligkeit der Szene angepasst.

Im Rahmen der geometrischen Aufbereitung sollen die Folgen einer eventuellen Fehlpositionierung des Satelliten korrigiert werden. Um die Szenen sinnvoll analysieren zu können, müssen sie korrekt und genau positioniert sein. Dies gilt insbesondere bei der Analyse einer Serie von Szenen derselben geographischen Region. Um eine normierte Positionierung der Szenen zu schaffen, werden aus jeder Szene, die einen Teil der zu analysierenden geographischen Region beinhaltet, quadratische Teile der Originalszene ausgeschnitten. Diese ausgeschnittenen Bereiche der ursprünglichen Szene werden Kacheln genannt. Dann wird jeder Pixel der Kachel auf die Zugehörigkeit zum Zielgebiet geprüft. Wenn ein Pixel relevant ist, wird er anhand seiner, aus der Position des Satelliten zum Aufnahmezeitpunkt ermittelten, Position in einem finalen Bild hinzugefügt. Dies garantiert eine einheitliche Grundlage für die Analyse der Zielregion.

Zusätzlich zu Fehlpositionierungen des Satelliten und den sich daraus ergebenden Abweichungen müssen möglicherweise noch weitere Störfaktoren durch die Aufbereitung gemindert werden. Hierzu zählen zum Beispiel durch die Atmosphäre verursachte Verschlechterungen, die aus Interferenzen innerhalb der Atmosphäre zwischen Erdoberfläche und dem Satelliten resultieren. Diese sollen korrigiert werden, um ein genaueres Satellitenbild zu erhalten. Techniken um diese Verbesserung zu erreichen sind beispielsweise das Strahlungstransfermodell, die bildbasierte atmosphärische Korrektur und die Histogramm-Minimum-Methode. Es ist individuell von der Szene und den zur

Verfügung stehenden Metadaten abhängig, mit welcher Methode die nützlichste Verbesserung erreicht werden kann.

Die Aufbereitung von Satellitenbildern muss vor einer wissenschaftlichen Analyse erfolgen, damit die Szenen unabhängig von Witterungseinflüssen, Atmosphäreninterferenzen, Fehlpositionierungen und sonstiger Störfaktoren untersucht werden können. Durch die zunehmend bessere Qualität von Satellitenbildern, die durch Fernerkundungssatelliten aufgezeichnet werden [MSWI04], können detailliertere Analysen getätigt werden. Jedoch steigt mit zunehmender Größe der Bilddateien auch der Rechenaufwand, um die Szenen aufzubereiten und zu analysieren. Mit zunehmender Datenmenge wird eine massiv parallelisierbare Vorgehensweise bei der Aufbereitung und der Analyse von Satellitenbildern attraktiver. Denn verteilte Systeme lassen sich meist kostengünstiger und flexibler erweitern als einzelne Maschinen, so dass das System bei einer unerwartet großen Datenmenge schnell erweitert werden kann. Dadurch lässt sich eine schnellere Ausführung der Prozesse erreichen.

## 2.2 Parallele Datenverarbeitungssysteme

Einleitung

### 2.2.1 Bedeutung und Eigenschaften von Big Data

Einleitung - Eigenschaften von Big Data

### 2.2.2 Verarbeitung großer Datenmengen - Konzepte und Systeme

Kurze Einleitung inkl. Überleitung zu Parallelität

Einleitung - Parallelität

Map Reduce - High Level Beschreibung + Funktionsweise, erklärt anhand Grafik + evtl. Beispiel.

Eigenschaften von Systemen (Hadoop, Spark, Storm, Asterix) -> Überleitung zu Flink

### 2.2.3 Apache Flink

Flink einführen + allgemein beschreiben

Konzept: 2nd order functions, UDFs, Operatoren + Tabelle

Konzept: JobManager: Masternode <-> Workernodes. Wie funktioniert die Parallelität?

Flink Optimierer: Eingabegraph, Jobgraph, Executiongraph

### 2.2.4 Python

Geschichte + grundlegende Sprachkonzepte



## Kapitel 3

# Beschreibung und Umsetzung des Algorithmus zur Analyse von Pixelzeitreihen

### 3.1 Beschreibung des Algorithmus zur Analyse von Pixelzeitreihen

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

### 3.2 Umsetzung des Algorithmus mit Apache Flink

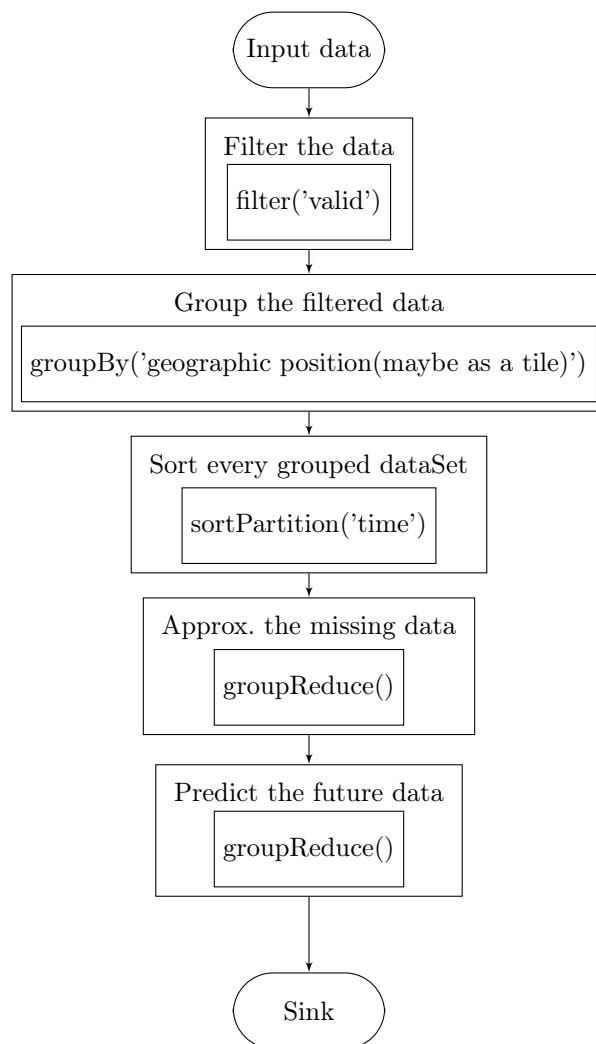
#### 3.2.1 Nutzung der Java-Programmierschnittstelle

#### 3.2.2 Nutzung der Python-Programmierschnittstelle

### 3.3 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

## Evaluierung

### 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

- [AAP<sup>+</sup>15] Shadab Alam, Franco D. Albareti, Carlos Allende Prieto, F. Anders, Scott F. Anderson, Timothy Anderton, Brett H. Andrews, Eric Armengaud, Áric Aubourg, Stephen Bailey, and et al. The eleventh AND twelfth data releases of the sloan digital sky survey: Final data from sdss-iii. *The Astrophysical Journal Supplement Series*, 219(1):12, Jul 2015.
- [Aba09] Daniel J Abadi. Data management in the cloud: Limitations and opportunities. *IEEE Data Eng. Bull.*, 32(1):3–12, 2009.
- [ABE<sup>+</sup>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.
- [AKL<sup>+</sup>12] Sattam Alsubaiee, Young-Seok Kim, Chen Li, Nicola Onose, Pouria Pirzadeh, Rares Vernica, Jian Wen, Yasser Altowim, Hotham Altwaijry, Alexander Behm, and et al. Asterix. *Proceedings of the VLDB Endowment*, 5(12):1898,1901, Aug 2012.
- [Bak10] Monya Baker. Next-generation sequencing: adjusting to data overload. *Nat Meth*, 7(7):495,Äi499, Jul 2010.
- [BEH<sup>+</sup>10] Dominic Battré, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl, and Daniel Warneke. Nephele/pacts: a programming model and execution framework for web-scale analytical processing. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 119–130. ACM, 2010.
- [Bel71] C Gordon Bell. C. mmp: the cmu multiminiprocessor computer, requirements and overview of the initial design. 1971.
- [BH85] A. Borodin and J.E. Hopcroft. Routing, merging, and sorting on parallel models of computation. *Journal of Computer and System Sciences*, 30(1):130,Äi145, Feb 1985.
- [Bre00] Eric Brewer. Towards robust distributed systems. 07 2000.
- [Cha11] Anju Chaudhary. Thermal infrared sensors. *Encyclopedia of Snow, Ice and Glaciers*, page 1156, 2011.
- [Deu12] Deutsches Institut für Normung e.V. *Photogrammetrie und Fernerkundung - Begriffe*, 8 2012. Rev. 3.
- [DG04] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. page 13, 2004.

- [DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce. *Communications of the ACM*, 51(1):107, Jan 2008.
- [EMC14] EMC<sup>2</sup>. The digital universe of opportunities. Technical report, EMC<sup>2</sup>, 2014.
- [Foua] Apache Software Foundation. Apache spark website. <https://spark.apache.org/>.
- [Foub] Apache Software Foundation. Apache storm website. <https://storm.apache.org/>.
- [Fouc] Apache Software Foundation. Flink website. <https://flink.apache.org/>.
- [Foud] Apache Software Foundation. Hadoop website. <https://hadoop.apache.org/>.
- [Fou15] Apache Software Foundation. The apache software foundation announces apache<sup>TM</sup> flink<sup>TM</sup> as a top-level project. [https://blogs.apache.org/foundation/entry/the\\\_apache\\\_software\\\_foundation\\\_announces69](https://blogs.apache.org/foundation/entry/the\_apache\_software\_foundation\_announces69), January 2015.
- [GP] GfZ-Potsdam. Geomultisens website. <http://www.geomultisens.gfz-potsdam.de/>.
- [HDF13] Kai Hwang, Jack Dongarra, and Geoffrey C Fox. *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.
- [HR83] Theo Haerder and Andreas Reuter. Principles of transaction-oriented database recovery. *ACM Computing Surveys (CSUR)*, 15(4):287–317, 1983.
- [Hü15] Fabian Hüske. Juggling with bits and bytes. <https://flink.apache.org/news/2015/05/11/Juggling-with-Bits-and-Bytes.html>, May 2015.
- [IDB12] James R. Irons, John L. Dwyer, and Julia A. Barsi. The next landsat satellite: The landsat data continuity mission. *Remote Sensing of Environment*, 122:11,21, Jul 2012.
- [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.
- [MH13] ABM Moniruzzaman and Syed Akhter Hossain. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*, 2013.
- [MSWI04] B.L. Markham, J.C. Storey, D.L. Williams, and J.R. Irons. Landsat sensor performance: history and current status. *IEEE Transactions on Geoscience and Remote Sensing*, 42(12):2691,2694, Dec 2004.
- [Sei85] Charles L Seitz. The cosmic cube. *Communications of the ACM*, 28(1):22–33, 1985.
- [TW12] Arthur Trew and Greg Wilson. *Past, present, parallel: a survey of available parallel computer systems*. Springer Science & Business Media, 2012.

- [WB72] William A Wulf and C Gordon Bell. C. mmp: a multi-mini-processor. In *Proceedings of the December 5-7, 1972, fall joint computer conference, part II*, pages 765–777. ACM, 1972.
- [WK09] Daniel Warneke and Odej Kao. Nephele: Efficient parallel data processing in the cloud. In *Proceedings of the 2Nd Workshop on Many-Task Computing on Grids and Supercomputers*, MTAGS '09, pages 8:1–8:10, New York, NY, USA, 2009. ACM.
- [YAJEA<sup>+</sup>00] Donald G. York, J. Adelman, Jr. John E. Anderson, Scott F. Anderson, James Annis, Neta A. Bahcall, J. A. Bakken, Robert Barkhouser, Steven Bastian, Eileen Berman, William N. Boroski, Steve Bracker, Charlie Briegel, John W. Briggs, J. Brinkmann, Robert Brunner, Scott Burles, Larry Carey, Michael A. Carr, Francisco J. Castander, Bing Chen, Patrick L. Colestock, A. J. Connolly, J. H. Crocker, Istvan Csabai, Paul C. Czarapata, John Eric Davis, Mamoru Doi, Tom Dombeck, Daniel Eisenstein, Nancy Ellman, Brian R. Elms, Michael L. Evans, Xiaohui Fan, Glenn R. Federwitz, Larry Fiscelli, Scott Friedman, Joshua A. Frieman, Masataka Fukugita, Bruce Gillespie, James E. Gunn, Vijay K. Gurbani, Ernst de Haas, Merle Haldeman, Frederick H. Harris, J. Hayes, Timothy M. Heckman, G. S. Hennessy, Robert B. Hindsley, Scott Holm, Donald J. Holmgren, Chi hao Huang, Charles Hull, Don Husby, Shin-Ichi Ichikawa, Takashi Ichikawa, Zeljko Ivezifá, Stephen Kent, Rita S. J. Kim, E. Kinney, Mark Klaene, A. N. Kleinman, S. Kleinman, G. R. Knapp, John Korienek, Richard G. Kron, Peter Z. Kunszt, D. Q. Lamb, B. Lee, R. French Leger, Siriluk Limmongkol, Carl Lindenmeyer, Daniel C. Long, Craig Loomis, Jon Loveday, Rich Lucinio, Robert H. Lupton, Bryan MacKinnon, Edward J. Mannery, P. M. Mantsch, Bruce Margon, Peregrine McGehee, Timothy A. McKay, Avery Meiksin, Aronne Merelli, David G. Monet, Jeffrey A. Munn, Vijay K. Narayanan, Thomas Nash, Eric Neilsen, Rich Neswold, Heidi Jo Newberg, R. C. Nichol, Tom Nicinski, Mario Nonino, Norio Okada, Sadanori Okamura, Jeremiah P. Ostriker, Russell Owen, A. George Pauls, John Peoples, R. L. Peterson, Donald Petravick, Jeffrey R. Pier, Adrian Pope, Ruth Pordes, Angela Prosapio, Ron Rechenmacher, Thomas R. Quinn, Gordon T. Richards, Michael W. Richmond, Claudio H. Rivetta, Constance M. Rockosi, Kurt Ruthmansdorfer, Dale Sandford, David J. Schlegel, Donald P. Schneider, Maki Sekiguchi, Gary Sergey, Kazuhiro Shimasaku, Walter A. Siegmund, Stephen Smee, J. Allyn Smith, S. Snedden, R. Stone, Chris Stoughton, Michael A. Strauss, Christopher Stubbs, Mark SubbaRao, Alexander S. Szalay, Istvan Szapudi, Gyula P. Szokoly, Anirudda R. Thakar, Christy Tremonti, Douglas L. Tucker, Alan Uomoto, Dan Vanden Berk, Michael S. Vogeley, Patrick Waddell, Shu i Wang, Masaru Watanabe, David H. Weinberg, Brian Yanny, and Naoki Yasuda. The sloan digital sky survey: Technical summary. *The Astronomical Journal*, 120(3):1579, 2000.
- [ZdP<sup>+</sup>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 15. Oktober 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, 15th October 2015

.....