

# Exposé für Studienarbeit: Vergleich von SciDB und Stratosphere zur Verarbeitung von Satellitenbildern

Moritz Borgmann

12. September 2014

BetreuerInnen Astrid Rheinländer  
Prof. Dr.-Ing. Ulf Leser  
Prof. Johann-Christoph Freytag

## 1 Motivation und Usecase

Sowohl für wissenschaftliche als auch für private Zwecke sind in den letzten Jahren immer größere und vollständigere Datensätze von Satellitenbildern zugänglich geworden. Die Größe dieser Datensätze führt zu neuen Problemen beim Speichern und Verarbeiten, insbesondere wenn Zugriffszeiten und Laufzeit bestimmter Algorithmen den Minuten- oder Stundenbereich nicht überschreiten sollen. Im Rahmen dieser Studienarbeit sollen anhand eines realen Usecases die Systeme SciDB<sup>1</sup> und Apache Flink, ehemals Stratosphere<sup>2</sup>, hinsichtlich ihrer Tauglichkeit zur Lösung des geschilderten Problems evaluiert werden.

Im Folgenden wird kurz die Herkunft der Satellitenbilder und die Arbeitsweise der Geographen mit diesen beschrieben. Darauf folgend wird der aus dem Arbeitsfluss der Geographie herausgelöste Usecase für diese Arbeit vorgestellt.

### 1.1 Satellitenbilder

Beim vorliegenden Problem geht es um die Verarbeitung von Satellitenbildern des Landsat-Programms. Die Satelliten werden von der NASA seit 1972

---

<sup>1</sup><http://www.scidb.org/>, abgerufen am 25.07.2014

<sup>2</sup><http://flink.incubator.apache.org>, abgerufen am 24.07.2014

zur Geofernerkundung eingesetzt und liefern Aufnahmen der Landoberfläche und Küsten. Auch wenn die eingesetzten Satelliten und Sensoren immer wieder technische Ausfälle hatten und die Daten aus der Anfangszeit sehr grob aufgelöst sind, so besteht ihr Vorteil in den langen Zeitreihen an Aufzeichnungen. Die gewonnenen Bilder finden vielfältige Anwendung: vom militärischen Einsatz (Aufklärung im Kalten Krieg), über wissenschaftliche Erkenntnisse (Entwicklung von Korallenriffen, antarktische Gletscher) bis zu privatwirtschaftlichen und privaten Anwendungen (Nachrichten und Google Maps). Bis heute werden, wenn auch inzwischen unter dem Namen LDCM (Landsat Data Continuity Mission), Datensätze gewonnen, die mit denen in den Jahren zuvor gewonnenen vergleichbar sind und insbesondere Aussagen über die Veränderungen der Umwelt im Laufe der Zeit ermöglichen<sup>3</sup>.

Der Orbit der Landsatsatelliten ist so gewählt, dass, je nach Satellit, alle 16-18 Tage das gleiche Gebiet beobachtet wird. Dabei findet die Aufnahme jeweils zur Uhrzeit der vorherigen Aufnahme statt. Es werden Streifen von etwa 185km Breite aufgenommen. Die Granularität der Aufnahmen unterscheidet sich je nach Satellit und Sensor. Während die ersten Satelliten Pixel lieferten, die 79 x 79m Landoberfläche darstellten, sind seit Landsat 7 im panchromatischen Kanal Pixel mit einer 15 x 15m Entsprechung möglich. Die Sensoren für sichtbares Licht (Blau, Grün, Rot) und im nahen und mittleren infraroten Bereich liefern seit 1982 Auflösungen mit 30 x 30m Pixeln.

Die Größe der Datensätze hängt ganz von gewähltem Sensor und Auflösung ab. Dabei erhält man die Daten in sogenannten „Szenen“, die einer Fläche von ca. 170km Nord-Süd x 183km Ost-West entsprechen. In den Archiven des USGS (United States Geographical Service) sind derzeit etwa 3 Millionen Szenen verfügbar, deren Größe zwischen 32 MB und 487 MB schwankt. Insgesamt sind ca. 1665 TB Rohdaten verfügbar. Während die Szenen früher einzeln bestellt und bezahlt werden mussten, sind sie heute beim USGS frei zum Download verfügbar. Während der Zugang zu den Daten also kein Hindernis mehr darstellt, ist das Problem nun die Menge der Daten zu beherrschen<sup>4,5</sup>.

## 1.2 Arbeitsweise der Geographie

Die Arbeit der Geographen mit den Satellitenbildern beginnt beim Herunterladen der Szenen vom USGS. Sobald sich diese auf den Servern des Instituts befinden, werden sie in zwei wesentlichen Schritten vorverarbeitet:

---

<sup>3</sup><http://de.wikipedia.org/wiki/Landsat>, abgerufen am 25.07.2014

<sup>4</sup><http://landsat.usgs.gov/USGSLandsatGlobalArchive.php>, abgerufen am 25.07.2014

<sup>5</sup><https://lpdaac.usgs.gov/>, abgerufen am 25.07.2014

- Korrektur atmosphärischer und optischer Störungen: Hier werden mithilfe weiterer Daten der NASA optische Verzerrungen durch Störungen in der Atmosphäre korrigiert.
- Erkennen und Maskieren von Wolken: Die Aufnahmen durchdringen Wolken nicht, sondern nehmen diese mit auf. Um die Ergebnisse nicht zu verfälschen, müssen sie erkannt und die Ergebnisse in einer Maske, der „CloudMask“ gespeichert werden. In allen weiteren Schritten kann darauf zugegriffen werden, um zu prüfen, ob ein Pixel Daten des Bodens enthält oder eine Wolke darstellt [Str05].

Nun liegen die Szenen prozessiert bereit und es wird ein Gebiet in Form eines Rechtecks definiert, für dessen Daten man sich interessiert. Die vorliegenden Daten sind szenenweise gespeichert. Wie die Szenen ein Gebiet abdecken, ist in Abbildung 1 dargestellt. Wie man sieht, sind die Szenen selbst nicht nach Nord und Süd ausgerichtet und überschneiden sich. In der Regel liegt ein solcher Datensatz ca. alle 14 Tage vor. Die Geographen definieren die obere linke (i.e. nordwestliche) Ecke des zu untersuchenden Rechtecks und dessen Maße in Breite und Länge. Nun müssen aus allen vorhandenen Daten jene Szenen ausgewählt werden, die Daten aus dem definierten Rechteck enthalten. Diese Daten müssen anschließend in ein gemeinsames File geschrieben werden.

Die so generierte Ausgabedatei kann dann mit weiterer Software von der Geographie analysiert und untersucht werden. Typische Fragestellungen sind dann etwa die Entwicklungen der Vegetation in einem bestimmtem Gebiet im Laufe der Zeit, Auswirkungen von Forstgesetzen [KMGR09] oder Diskrepanzen zwischen staatlichen Angaben und tatsächlichen Zuständen des Waldbestandes [VBL<sup>+</sup>12].

### 1.3 Usecase

Die Auswahl des zu implementierenden Teilproblems wurde im Wesentlichen so gewählt, dass nach Abschluss der Programmierung ein funktionierender Prototyp entsteht, mit dem die Geographen bereits arbeiten können. Dadurch wäre es im Falle einer Vertiefung der Arbeit möglich, auf dem implementierten System aufzubauen und Nutzerfeedback einfließen zu lassen.

Der Usecase, der für die Evaluierung exemplarisch umgesetzt werden soll, beschäftigt sich mit der Auswahl der relevanten Daten im Untersuchungsgebiet. Die Menge der vorhandenen Daten respektive Satellitenbildern deckt in der Regel größere Gebiete ab, als sie für die Geographen von Interesse sind.

---

<sup>6</sup><http://www.tnris.org/new-landsat-satellite-set-to-launch>, abgerufen am 10.08.2014

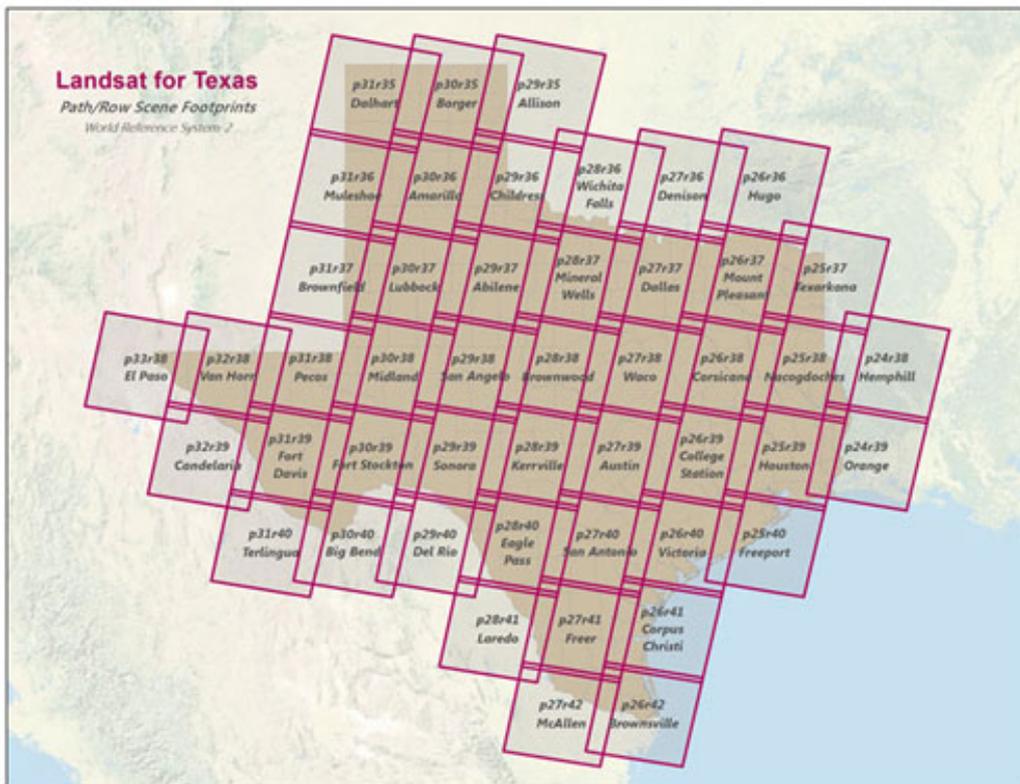


Abbildung 1: Anordnung der Landsat-Szenen über Texas<sup>6</sup>.

Diese definieren, wie bereits erwähnt, ein Gebiet in Form eines Rechteckes, für das sie sich interessieren und welches durch die obere rechte (nordwestliche) Ecke, sowie die Ausmaße definiert wird. Um die Daten dieses Gebiets weiter zu verarbeiten, müssen aus der Gesamtheit der vorhandenen Daten jene ausgewählt werden, die innerhalb dieses Rechteckes liegen. Diese Daten werden dann in eine einzelne Datei geschrieben, um sie für die Werkzeuge in den nachfolgenden Untersuchungen gut beherrschbar zu machen. Eine Auswahl nach bestimmten Bändern oder Zeiträumen wäre auch denkbar, wenn sich dies leicht integrieren ließe.

Die Entscheidung, ob Daten für das Rechteck relevant sind oder nicht, kann anhand der geographischen Koordinaten des Szenen bestimmt werden. Sobald eine Szene auch nur einen kleinen Teil des Rechtecks abdeckt, sollen die kompletten Daten dieser Szene in der Ausgabedatei enthalten sein.

Die beide Implementierung, also sowohl SciDB als auch Apache Flink, sollten also folgendes leisten:

- Einlesen und Speichern der Datensätze, sowohl der eigentlichen Daten

als auch dazu gehöriger Metadaten.

- Wenn nötig, das Anlegen von Zugriffsstrukturen.
- Verarbeiten einfacher Anfragen im Stile von „Gib mir alle Daten aus dem Bereich von (x1, y1) bis (x2, y2)“.
- Ausgabe der Anfrageergebnisse in einem definierten Format.

Der gewählte Usecase passt recht gut zu den aufgestellten Anforderungen. Auch in den vorhandenen Arbeitsabläufen der Geographen ist der Usecase ein einzelner Arbeitsschritt, der unabhängig von vorhergehenden und nachgelagerten Prozessen ist und mit fest definierten Eingabe- und Ausgabeformaten arbeitet. Wenn im Rahmen der Arbeit eine wesentlich schnellere Lösung als die existierende gebaut werden kann, ist es nicht unwahrscheinlich, dass sie dauerhafte Anwendung in der Geographie findet.

## 2 Zielstellung

Das Ziel dieser Arbeit besteht darin, Stratosphere und SciDB im Hinblick auf ihre Fähigkeiten für die Arbeit mit Satellitenbildern zu vergleichen. Hierzu soll der Workflow der Anwender dokumentiert und analysiert werden, um deren Bedürfnisse und Anforderungen an ein solches System zu identifizieren und gewichten. Anschließend soll der Usecase der Datenauswahl in beiden Systemen umgesetzt werden, um die Machbarkeit größerer Anwendungsfälle oder der kompletten Abbildung des Workflows in den jeweiligen Systemen hinsichtlich Aufwandes und Qualität zu vergleichen.

Neben den funktionalen Anforderungen soll auch untersucht werden, ob bestimmte qualitative Anforderungen bestehen. Nachdem die Anforderungen und Qualitätskriterien aus Sicht der Anwender identifiziert wurden, können die beiden Systeme bezüglich dieser Kriterien untersucht werden. Werden bestimmte Anforderungen schon vom System an sich unterstützt? Was muss selber implementiert werden? Wie leicht sind bestimmte Qualitätskriterien zu erreichen; was wird voraussichtlich zu Problemen führen?

Mögliche Kriterien wären:

- Einarbeitungszeit in das jeweilige System
- Aufwand der Implementierung
- Laufzeit der jeweiligen Implementierung von Fragestellung bis Lösung

- Aufwand und Laufzeit, um neue Daten in das jeweilige System aufzunehmen
- Qualität von Dokumentation und Community bei Problemen

Nachdem beide Implementierungen umgesetzt und getestet sind, können sie hinsichtlich der zuvor erarbeiteten Qualitäts- und Akzeptanzkriterien überprüft werden. Sofern diese quantitativ erfassbar sind, ist ein Vergleich relativ leicht. Hinsichtlich einiger objektiver Kriterien (i.e. Wartbarkeit oder Erweiterbarkeit) ist eine ausführliche Auswertung angebracht. Sicherlich können nicht alle Kriterien untersucht werden, jedoch möchte ich die vermutlich einflussreichsten genauer untersuchen und aufzeigen, welche Probleme im jeweiligen System bei weiterem Ausbau zu erwarten sind.

Ob eine Empfehlung oder ein Abraten bezüglich eines der beiden Systeme möglich sein wird, ist noch nicht abzuschätzen.

Nachdem das Teilproblem implementiert wurde, muss es beim Anwender vorgestellt und getestet werden. Welchen Anforderungen hinsichtlich Funktionalität und Qualität wird die jeweilige Lösung gerecht? Welche Anforderungen konnten nicht mit vertretbarem Aufwand erreicht werden oder können eventuell überhaupt nicht umgesetzt werden? Sind es Probleme, die nur einmalig bei der Anpassung der Systeme an den neuen Kontext auftreten oder benötigen diese Probleme auch bei der alltäglichen Arbeit Beachtung?

Welche Laufzeiten werden erreicht? Welche Erfahrung aus der Entwicklung hinsichtlich weicher Kriterien wie Dokumentation, Community, Support und Zugänglichkeit wurden gemacht? Wie hoch ist der Aufwand, um bestimmte Fragestellungen zu programmieren?

In dieser Betrachtung sollten alle wesentlichen Punkte, die wichtig sind, um zu entscheiden mit welchem System man weiter arbeiten möchte, abgedeckt sein.

### 3 Zu untersuchende Systeme

Im Folgenden stelle ich die beiden Systeme Apache Flink und SciDB kurz vor. Da ein Anwender-Usecase untersucht wird, betrachte ich die Features der jeweiligen Systeme aus Sicht eines Anwenders. Die interne Verarbeitung der Systeme und die Speicherung der Daten sind dabei für die Anwender zunächst unwichtig, da sie die Systeme nutzen, aber nicht massiv ändern oder erweitern wollen. Beide Systeme haben das Ziel Anfragen an große Datenmengen möglichst schnell zu verarbeiten und dem Anwender auf einfache Art und Weise Schnittstellen zum Zugriff und Bearbeiten der Daten bereitzustellen.

### 3.1 Apache Flink

An Apache Flink wird unter dem Namen „Stratosphere“ seit 2009 mit Unterstützung der DFG (Deutsche Forschungsgemeinschaft) vor allem in Berlin gearbeitet. Inzwischen wird das Projekt durch Apache unterstützt, wo es unter dem Namen „Flink“ geführt wird. Es ist ein Open Source-Projekt, das auf HDFS und YARN aufsetzt, mit dem Ziel, ein hohes Level an Abstraktion für die Lösung von BigData-Problemen bereitzustellen. Intern wird mit einem Key-Value Pattern gearbeitet. Apache Flink bietet eine API nach Java und Scala, aber auch eine eigene Skriptsprache namens „Meteor“. Eine Python API ist in Entwicklung. Durch das Bereitstellen von Funktionen, die über Map/Reduce hinausgehen, können viele typische Aufgaben wie Iterationen, Join oder Cross abstrahiert werden. So ist es möglich, auch komplexere Anfragen und Programme zu formulieren und durch Apache Flink ausführen zu lassen.

Das System bietet mehrere Abstraktionsschichten, wobei dem Anwender frei gestellt ist, auf welcher er arbeiten möchte. So kann mit der deskriptiven Sprache Meteor sehr abstrakt gearbeitet werden oder direkt in Java mit einem nicht viel höherem Abstraktionsniveau als bei klassischem MapReduce [ABE<sup>+</sup>14].

Apache Flink funktioniert und ist momentan stark in Entwicklung, größere Anwendungen im Produktionsbetrieb sind jedoch noch nicht bekannt.

In Listing 1 ist ein in Scala geschriebenes Wordcount-Beispiel zu sehen. Das bedeutet, dass eine Eingabedatei geöffnet und in einzelne Wörter zerlegt wird. In diesem Fall beginnt ein neues Wort immer nach einem Leerzeichen. Im nächsten Schritt wird dann gezählt, wie oft jedes Wort vorgekommen ist. Das Ergebnis wird in eine CSV-Datei geschrieben<sup>7</sup>.

```
1 // read input data
2 val input = TextFile(textInput)
3
4 // tokenize words
5 val words = input.flatMap { _.split(" ") map { (_, 1) } }
6
7 // count by word
8 val counts = words.groupBy { case (word, _) => word }
9   .reduce { (w1, w2) => (w1._1, w1._2 + w2._2) }
10
11 val output = counts.write(wordsOutput, CsvOutputFormat())
```

Listing 1: Beispiel Wordcount in Scala mit Stratosphere

---

<sup>7</sup>[http://flink.incubator.apache.org/docs/0.6-SNAPSHOT/scala\\_api\\_examples.html#word-count](http://flink.incubator.apache.org/docs/0.6-SNAPSHOT/scala_api_examples.html#word-count), abgerufen am 25.07.2014

## 3.2 SciDB

Die Entwicklung von SciDB begann, nachdem aus der Wissenschaft der Wunsch nach Datenbanksystemen aufkam, die für wissenschaftliche Probleme das darstellen, was RDBMs für wirtschaftliche Probleme darstellen. Die Art der Anforderungen und Daten sind in beiden Bereichen sehr unterschiedlich.

Unter der Regie von Michael Stonebraker und mit kommerzieller Unterstützung wurde 2008 die erste Version von SciDB veröffentlicht; um diesen Anforderungen gerecht zu werden. SciDB ist in C++ geschrieben und Open Source, was ein wichtiges Kriterium für wissenschaftliche Anwendungen darstellt.

Das System arbeitet nicht auf Tabellen wie RDBMs sondern auf Arrays. Dies hat mehrere Vorteile. Beispielsweise liegen Daten aus wissenschaftlichen Feldern mit großen Datenmengen (Geospatial, Simulationen, Genome) in der Regel bereits als Array vor. So werden auch die Landsatbilder als Array vom USGS bezogen. Die Metadaten zu den eigentlichen Bildern wiederum bestehen aus Textfiles, jedoch wird auch das Speichern von Metadaten von SciDB bereits unterstützt[SBPR11].

Als besondere Features des Systems sind noch zu nennen<sup>8</sup>:

- Versionierung der Daten um Nachvollziehbarkeit von Ergebnissen zu gewährleisten
- Verschiedene Operatoren and Analysen der linearen Algebra werden nativ unterstützt
- Eventuelle Unsicherheiten der Daten, wie sie etwa bei Messungen auftreten, können modelliert werden.

Auch wenn SciDB auf den ersten Blick vermuten lässt, dass es bereits viele benötigte Features für Geodaten liefert, bleibt die Frage, wie viel Aufwand für spezielle Änderungen nötig ist und ob solche überhaupt möglich wären.

Das System kann mit einer SQL-ähnlichen Sprache, der AQL (Array Query Language), sowie mit einer funktionalen Sprache, der AFL (Array Functional Language), gesteuert werden. Außerdem ist es möglich, das System um eigene Algorithmen zu erweitern, um die Bearbeitung von Fragestellungen zu beschleunigen („move the computation to the data“).

In Listing 2 sind Beispiele für AQL und AFL. Zunächst wird in AFL ein Array A gefiltert und anschließend mit einem Array B gejoint. Im AQL Beispiel ist das Vorbild SQL gleich zu erkennen. Auch hier wird Array A gefiltert und mit Array B gejoint.

---

<sup>8</sup><http://en.wikipedia.org/wiki/SciDB>, abgerufen am 25.07.2014

```

1 AFL:
2   temp = filter (A, c = values)
3   result = join (B, temp: I, J)
4
5 AQL:
6   select * from A, B where
7     A.I = B.I and
8     A.J = B.J and
9     A.c = value

```

Listing 2: Beispiel für AFL und AQL: Zwei Arrays A und B mit Dimensionen I und J. A hat das Attribut C [SBPR11].

## 4 Arbeitspakete

Die im Folgenden geschilderten Arbeitspakete fallen in der Regel für beide Systeme an. Das heißt, eine Analyse des Systems muss sowohl für Stratosphere als auch für SciDB erfolgen, ebenso wie Implementierung und Evaluierung. Diese Arbeitsschritte sollten bei beiden System gleich sein. Ob die Schritte in beiden Systemen parallel oder alle Schritte erst in einem und dann im anderen umgesetzt werden, möchte ich erst nach genauer Evaluierung beider Systeme entscheiden.

- Einarbeitungszeit in das jeweilige System
- Aufwand der Implementierung
- Laufzeit der jeweiligen Implementierung von Fragestellung bis Lösung
- Aufwand und Laufzeit, um neue Daten in das jeweilige System aufzunehmen
- Qualität von Dokumentation und Community bei Problemen

### 4.1 Analyse

Bevor ich mit der Implementierung in den jeweiligen Systemen beginne, sollen diese zunächst hinsichtlich ihrer Fähigkeiten in Bezug auf den Usecase untersucht werden. Insbesondere soll überprüft werden, welche Anforderungen sowohl funktionaler als auch qualitativer Art vom jeweiligen System vielleicht schon unterstützt oder gar als Feature dort implementiert sind. So soll unnötiger Aufwand vermieden werden.

## 4.2 Implementierung Speicherung

Ist die Analyse fertig, wird die eigentliche Umsetzung begonnen. Das bedeutet, dass die Systeme zuerst aufgesetzt und installiert werden müssen. Anschließend muss überlegt werden, wie die Daten für das bzw. in dem jeweiligen System möglichst gut abgespeichert werden, um schnellen Zugriff und Verarbeitung zu gewährleisten. Hier wird sich schon abzeichnen, ob eine ausreichende Geschwindigkeit erreicht werden kann, oder ob weitere Zugriffsstrukturen notwendig sind.

## 4.3 Implementierung Workflow

Nun wird eine Schnittstelle für die Nutzer implementiert, über die sie die gewünschten Daten definieren. Den Fähigkeiten der Anwender entsprechend muss dies keine grafische Oberfläche sein; Bedienung im Terminal reicht aus. Abschließend kann die Logik implementiert werden, um die Daten zu finden und im gewünschten Format auszugeben.

## 4.4 Evaluierung

Eventuell ist mit den Anwendern bezüglich der Implementationen zu halten, um kleinere Änderungswünsche zu realisieren.

Nach der Implementierung erfolgt die Evaluierung. Das bedeutet in diesem Fall:

- Konnten die funktionalen Anforderungen der Anwender erreicht werden?
- Konnten qualitative Anforderungen der Anwender erreicht werden?
- Wie hoch war der Aufwand sich in die jeweiligen Systeme einzuarbeiten?
- Wie hoch ist der Aufwand, die jeweiligen Systeme an die Arbeit mit Satellitenbildern anzupassen?
- Welche Laufzeiten können erreicht werden?
- Parallelisierbarkeit und Skalierbarkeit, insbesondere horizontal.
- Welche der beiden Lösungen wird von den Anwendern bevorzugt?

Es wird ein Fazit der jeweiligen Systeme gezogen bezüglich der Stärken und Schwächen bei der Verarbeitung von Satellitendaten.

Abschließend wird versucht, eine Empfehlung abzugeben, mit welchem System der weitere Workflow abgebildet werden sollte.

## Literatur

- [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, et al. The stratosphere platform for big data analytics. *The VLDB Journal*, pages 1–26, 2014.
- [KMGR09] Tobias Kuemmerle, Daniel Müller, Patrick Griffiths, and Marioara Rusu. Land use change in southern romania after the collapse of socialism. *Regional Environmental Change*, 9(1):1–12, 2009.
- [SBPR11] Michael Stonebraker, Paul Brown, Alex Poliakov, and Suchi Raman. The architecture of scidb. In *Scientific and Statistical Database Management*, pages 1–16. Springer, 2011.
- [Str05] Kathleen I Strabala. *MODIS cloud mask users guide*. University of Wisconsin–Madison, 2005.
- [VBL<sup>+</sup>12] Veerle Vanacker, Vincent Balthazar, Eric Lambin, Jaclyn Hall, Patrick Hostert, Patrick Griffiths, Steven Vanonckelen, Armando Molina, and Anton Van Rompaey. Forest transition in mountain environments: topographic corrections and modelling of ecosystem services. 2012.