

# **Arbeitstitel: Analyse Wissenschaftlicher Daten mit BigData Werkzeugen**

**— Exposé —**

Arbeitsbereich Wissenschaftliches Rechnen  
Fachbereich Informatik  
Fakultät für Mathematik, Informatik und Naturwissenschaften  
Universität Hamburg

Vorgelegt von:  
E-Mail-Adresse:  
Studiengang:

Arne Struck  
1struck@informatik.uni-hamburg.de  
Bsc. Informatik

Betreuer:

Julian Kunkel

Hamburg, den 21. Juni 2015



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Motivation . . . . .	5
1.2	Zielsetzung . . . . .	7
<b>2</b>	<b>Problemstellung</b>	<b>8</b>
<b>3</b>	<b>Theoretischer Hintergrund</b>	<b>9</b>
<b>4</b>	<b>Methodik und Vorgehensweise</b>	<b>11</b>
4.1	Design . . . . .	11
4.2	Implementation und Verlauf . . . . .	11
4.3	Ergebnisevaluation . . . . .	12
<b>5</b>	<b>Zeitplan</b>	<b>13</b>
<b>6</b>	<b>Vorschlag Struktur der Arbeit</b>	<b>14</b>
	Möglicher Ausblick	
	Literaturverzeichnis	



# 1 Einleitung

## 1.1 Motivation

Der Bereich des High Performance Computing ist seit Jahrzehnten ein wichtiges Forschungs- und Anwendungsgebiet der Informatik. Oftmals befasst sich das High Performance Computing mit Fragen der Meteorologie, Klimatologie, theoretischen Physik und Biologie. Die Lösungen dieser Fragen wird durch die Simulation von Modellen aus dem entsprechenden Bereich und die Analyse von gemessenen und berechneten Daten approximiert.

Dies sind meist sehr rechenintensive Tätigkeiten, daher sind extrem leistungsstarke Rechner von Nöten. Heutzutage werden zu diesem Zweck meistens Cluster, also verteilte Rechensysteme eingesetzt. Bei der Realisierung der oben erwähnten Tätigkeiten auf verteilten Systemen ist die Frage der Parallelisierung von Berechnungen eine entscheidende. Im High Performance Computing Bereich wird dies seit Jahrzehnten großteilig durch Message Passing Interface MPI (beispielsweise implementiert durch Open MPI) erreicht [Lud14].

Durch das Aufkommen des Internets als Massenmedium in den letzten beiden Jahrzehnten hat sich eine Ökonomie entwickelt, welche große Datenmengen managen und verarbeiten muss. Auch diese Ökonomie, deren Geschäftsfeld allgemein als Big Data bezeichnet wird, setzt stark auf verteilte Systeme.

Allerdings wird hier nicht auf MPI für das Arbeiten auf verteilten Systemen gesetzt, sondern andere Lösungen für die Problematiken, welche verteilte Berechnungen mit sich bringen, bevorzugt. Als Beispiel für Komplettlösungen wären spark und hadoop zu nennen, welche nicht auf dem MPI-Technologie Stack aufbauen. Allerdings existieren auch Werkzeuge, welche sich in die bisherigen Systeme leichter integrieren lassen, wie das Data Base Management System Rasdaman.

Diese Herangehensweisen versprechen leichtere Nutzbarkeit und Umsetzbarkeit durch Reduktion des durch die Parallelisierung bedingten Overheads im Programm, sowie höhere Flexibilität des Codes. Analysen der Interessenlagen bezüglich der verschiedenen Herangehensweisen weisen auf einen Trend in Richtung der Lösungen der Big Data Unternehmen hin, wie in Abbildung 1.1 und 1.2 zu sehen. Die "Big Data Unternehmen" besitzen weiterhin das Potential High Performance Computing in Größenordnungen anzubieten, welche durchaus die der HPC Community übersteigen (erste Tendenzen in diese Richtung sind momentan zu beobachten [Lud14]).

Wegen der Verschiebung der Interessenlagen, der relativen Schwerfälligkeit MPIs und nicht zuletzt der Übermacht der Privatwirtschaft drängen sich für das klassische HPC die Frage auf, beziehungsweise ob die Lösungen aus der Big Data Ökonomie leistungstechnisch mit MPI konkurrieren können und somit eine zumindest partielle Alternative auch für die

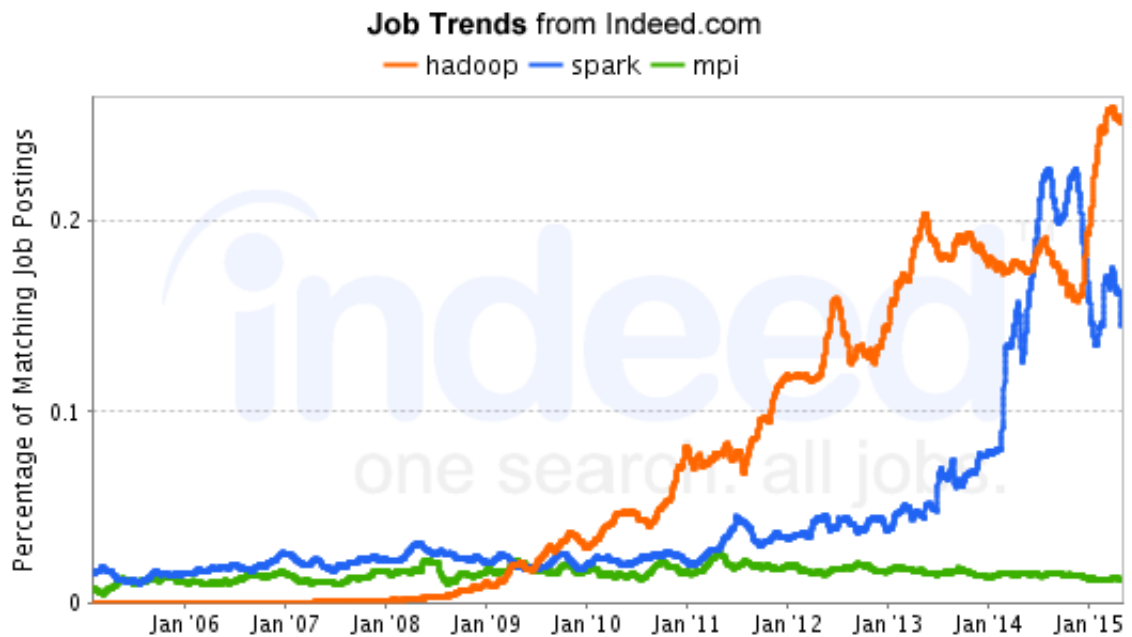


Abbildung 1.1: Jobangebote im betrachteten Bereich

momentane High Performance Computing Community darstellen.

Die momentane Messungslage deutet darauf hin, dass die Werkzeuge aus der Big Data Industrie für die Berechnung von Simulationen nicht ausreicht [Lud14]. Allerdings ist es möglich, dass ihre Leistungsfähigkeit für die Nachbereitung und Analysetätigkeiten im HPC-Bereich ausreichend ist.

hier kurze Beschreibung gestalten der Nachbereitung gestalten?

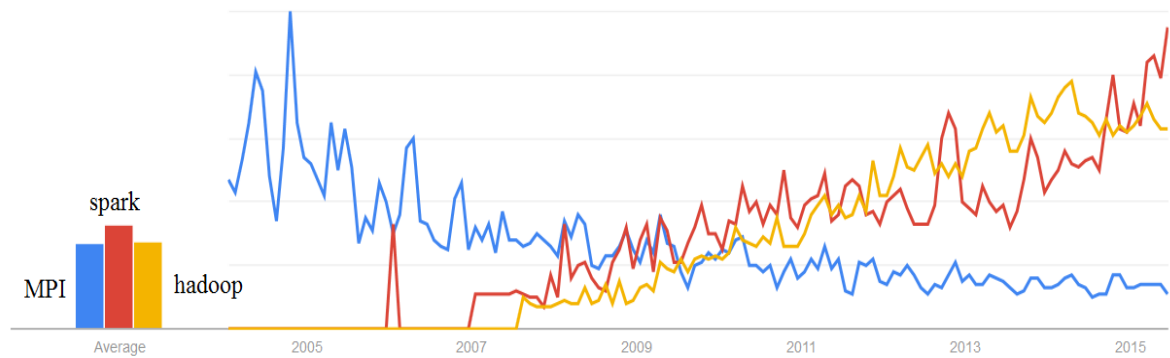


Abbildung 1.2: Google Trends: MPI, Spark, Hadoop

## 1.2 Zielsetzung

Diese Arbeit soll nun die zuvor erwähnte Eignung einiger "Big Data" Werkzeuge im Bezug auf die Nachbereitung wissenschaftlicher Daten analysieren. Hierzu muss sowohl ein Vergleich untereinander, als auch zu den bisherigen Verfahrensweisen gezogen werden. Für einen aussagekräftigen Vergleich für die Nutzbarkeit sind sowohl eine Bewertung des Bedienkomforts, als auch ein Vergleich der jeweiligen Einarbeitungszeiten und Nutzungszeiten.

## 2 Problemstellung

Um das in 1.2 definierte Ziel zu erreichen müssen viele Problematiken gelöst werden. Da die Palette der in Frage kommenden Werkzeuge nahezu keine Grenzen kennt, muss eine Auswahl orientiert am begrenzten Zeitraum der Arbeit getroffen werden. Da sowohl eine Einarbeitungszeit in die jeweiligen Werkzeuge, als auch eine Umsetzung für jedes Werkzeug von Nöten sind, aber auch eine gewisse Varianz von Nöten ist, wären 3 Werkzeuge eine naheliegende Wahl. Um die Integrierbarkeit in andere Systeme zu gewährleisten und die Einarbeitungszeit nicht durch das Aufsetzen ganzer Systeme zu beschränken, sollte die Wahl auf nicht Komplettlösungen eingeschränkt werden.

Ein weiteres Teilproblem der Eignungsbewertung stellt der Entwurf einheitlicher, geeigneter Testdaten dar. Hierzu sollten die Daten sowohl eine ausreichende Größe besitzen, als auch aus dem High Performance Computing stammen.

Des weiteren muss eine Herangehensweise zur eigentlichen Eignungsanalyse gestaltet werden. Dieses Teilproblem gliedert sich in zwei Problemtteile, zum einen die Definition der Bewertungskriterien, zum anderen Definitionen standardisierter Beispielfunktionalitäten und Herangehensweisen an die Messungen.

unsicher, ob im Sinne des Erfinders



### 3 Theoretischer Hintergrund

Im HPC-Bereich werden heutzutage große, berechnungsintensive Modelle berechnet. Im wissenschaftlichen Bereich heißt dies, dass es sich um Modelle komplexer Systeme handelt. Ein Beispiel wäre hier ein Klimamodell, wobei Einzelheiten wie Wolkensysteme bis hin zu Verdunstungsmengen im Meer berechnet werden. Dies ist nötig, da sich bei solchen Systemen, wie dem Klima alle Teilsysteme sich gegenseitig beeinflussen und das Einführen von nicht berechneten Einflüssen durch Konstanten oder das nicht beachten selbiger Einflüsse zu den Ungenauigkeiten der Berechnungen beiträgt.

Dies bedeutet, dass die Ergebnisdaten aus vielen verschiedenen Daten in großen Mengen besteht, von denen für die Meisten Fälle nicht alle relevant sind. Für das Beispiel des Klimamodells bedeutet dies, dass die meisten berechneten Daten für jemanden, der sich beispielsweise für die Entwicklung von Temperaturdaten nicht von Nöten sind. Da dies zweifellos nicht die einzelne Anwendungsmöglichkeit darstellt, lohnt es sich allerdings die anderen Daten zu berechnen. Hier setzt die Nachbereitung ein.

Diese Datenmengen bedeuten allerdings, dass auch die Nachbereitung ein rechenintensiver Prozess ist. Bei der Nachbereitung handelt es sich nicht nur um das einfache Abfragen oder Filtern aus dem Datenbestand, sondern um mögliche weitere Berechnungen mithilfe der Daten, das Erstellen von Vergleichsstatistiken oder Visualisierungen. Außer im Fall der weiteren Berechnungen, welche parallelisiert werden könnten, sind die meisten Standardvorgehen selbst keine High-Performance-Applications [AKX<sup>+</sup>13]. Des weiteren werden öfters Workarounds für größere Datenmengen gebraucht und für größere, in einem Grid organisierten Datenmengen, welche im HPC Bereich üblich sind, siehe 3.1. Tatsächlich kann dieser Teil der Nachbereitung bereits bei der Klimaberechnung ein

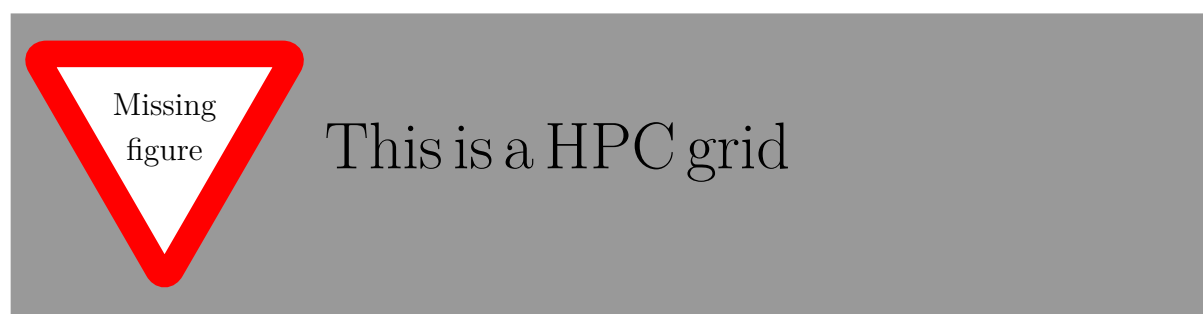


Abbildung 3.1: Grid Layout in HPC

Bottle-Neck darstellen [Lud14] [AKX<sup>+</sup>13].

- noch etwas über CDO (Climate Data Operators), gerade am DKRZ genutzt (commandozeilen-tool für den zweck)?
- Datenaufteilung vs Codeaufteilung (datenaufteilung relevant, weil grids erwähnt)
- wobei: wie viel grundlagen für hpc?

## 4 Methodik und Vorgehensweise

In diesem Kapitel wird ein Vorgehen bei der Lösung der in 2 beschriebenen Problemstellung entworfen.

### 4.1 Design

Nach einer Entscheidung für die Werkzeuge, müssen zu erst die Testdaten zusammengestellt werden. Es werden größere Datenmengen benötigt werden, um die Leistungsstärke der Werkzeuge sachgemäß bewerten zu können. Allerdings werden auch verschiedene Daten für die Testdaten benötigt.

Der nächste Schritt stellt die Aufstellung geeigneter Vergleichsmetriken dar, welche in den verschiedenen Werkzeugen implementiert werden müssen.

Für Implementationszeit und Berechnungszeit ist die Vergleichsmetrik offensichtlich. Dagegen muss für den Benutzerkomfort eine Vergleichsmetrik entworfen werden, hierbei sollten die Einfachheit der Bedienung, die Einarbeitungszeit, Installationszeit ihre Rollen spielen. Natürlich handelt es sich hierbei um partiell subjektive Bewertungen, worauf allerdings viele schlecht diskretisierbaren Probleme hinauslaufen. Sie müssen allerdings als solche gekennzeichnet werden.

### 4.2 Implementation und Verlauf

Zu erst müssen die festgelegten Vergleichsmetriken in den verschiedenen Werkzeugen implementiert werden. Diese bestehen zum einen aus verschiedenen exemplarischen, in dieser Form auftretender Queries auf die Daten, weiteres Filtern mit anschließender Interpolation und, oder Darstellung der Ergebnisse. Hierbei müssen sowohl die Implementationszeit, der Benutzerkomfort, als auch die Berechnungszeit gemessen werden.

Nachdem die Tests auf den standardisierten Testdaten durchgeführt wurden, sollen im Anschluss die Testdaten auf die Anforderungen der Werkzeuge optimiert werden. Hierauf folgt ein zweiter Berechnungsdurchlauf, welcher zeigen soll ob eine abgestimmte Modifikation der Testdaten auf das jeweilige Werkzeug eine Leistungsverbesserung ergibt, wobei auch die Modifikationszeit berücksichtigt werden muss. Dies sollte auch einen Einfluss auf die Bewertung des Benutzerkomforts haben, wenn die Optimierung von vorhandenen Datensätzen für den Nutzer unangenehme Mengen an Mehraufwand mit sich bringt.

Alle Berechnungen müssen mehrmals durchgeführt werden, um dem Einfluss von Outliern bei den Messungen vorzubeugen.

## 4.3 Ergebnisevaluation

Nun müssen die verschiedenen Messergebnisse verglichen werden, sowohl mit denen von CDOs, als auch mit denen der gewählten Werkzeuge. Die Messungsergebnisse dürfen in den Zeitkomponenten nicht die von CDO überschreiten, um das Werkzeug als geeignet zu bewerten.

Des weiteren muss bewertet werden für welche Anwendungsgebiete und Zwecke das entsprechende Werkzeug geeignet ist, beispielsweise die Eignung für interaktives Arbeiten und oder interaktive Darstellung nach der Bearbeitung. Die Bewertung hierzu sollte in Relation zu den Möglichkeiten der bisherigen Verfahrensweisen stehen.

## 5 Zeitplan

Phase	Gegenstand	veranschlagte Zeit
Recherche	Literaturrecherche Theorieteil der Arbeit	1 Woche
Design	Theoretische Umsetzung	2 Wochen
Werkzeug 1	Implementation, Datenerhebung, mit Werkzeug 1	2 Wochen
Werkzeug 2	Implementation, Datenerhebung, mit Werkzeug 2	2 Wochen
Werkzeug 3	Implementation, Datenerhebung, mit Werkzeug 3	2 Wochen
Evaluation	Analyse, Interpretation	1 Woche
Abschluss	Schluss schreiben und Überarbeitung	1 Woche
Restzeit	Pufferzone für Phasen, die länger dauern	1 Woche

Tabelle 5.1: Zeitplan

Bearbeitungsdauer widerspiegelt, nicht der Bearbeitungszeitraum

Restzeit ist für Arbeiten vorgesehen, bei denen sich herauskristalisiert, dass die veranschlagte Zeit nicht genügt.

# 6 Vorschlag Struktur der Arbeit

## 1 Einleitung

- 1.1 Motivation . . . . .
- 1.2 Zielsetzung . . . . .

## 2 Grundlagen

- 2.1 Beschreibung Werkzeug 1 . . . . .

wahrscheinlich R mit Big Data Erweiterungen

- 2.2 Beschreibung Werkzeug 2 . . . . .

wahrscheinlich SciDB

- 2.3 Beschreibung Werkzeug 3 . . . . .

wahrscheinlich Rasdaman

## 3 Anforderungen

## 4 Design

- 4.1 Entwurf Bewertungskriterien . . . . .
- 4.2 Entwurf Funktionalitäten . . . . .
- 4.3 Entwurf Testdatensätze . . . . .

## 5 Implementation

- 5.1 Implementationen der Funktionalitäten . . . . .
- 5.2 Optimierungen der Testdaten . . . . .

## 6 Evaluation

- 6.1 Vergleich der gemessenen Daten . . . . .
- 6.2 Vergleich der gemessenen Daten (optimierte Datenspeicherung) . . . . .
- 6.3 Interpretation möglicher Anwendungsfälle . . . . .

## 7 Schluss

- 7.1 Zusammenfassung . . . . .
- 7.2 Fazit . . . . .
- 7.3 Ausblick . . . . .

## 7 Möglicher Ausblick

Die gestalteten und implementierten Funktionalitäten könnten die Grundlage einer Erweiterung für wissenschaftliches Rechnen für das am besten geeignete Werkzeug darstellen. Des weiteren wird, je nach Ergebnis der Arbeit unter Umständen ein erweiterter Praxistest des entsprechenden Werkzeugs von Nöten sein.

Erweitern

# Literaturverzeichnis

- [AKX<sup>+</sup>13] Vassil Alexandrov, Jayesh Krishna, Xiabing Xu, Tim Tautges, Iulian Grindeanu, Rob Latham, Kara Peterson, Pavel Bochev, Mary Haley, David Brown, Richard Brownrigg, Dennis Shea, Wei Huang, and Don Middleton. Parncl and pargal: Data-parallel tools for postprocessing of large-scale earth science data. In *International Conference on Computational Science, ICCS 2013*. Elsevier, 2013.
- [Dur] Jonathan Dursi. Hpc is dying, and mpi is killing it. <http://www.dursi.ca/hpc-is-dying-and-mpi-is-killing-it/>.
- [Kun13] Julian Martin Kunkel. *Simulation of Parallel Programs on Application and System Level*. PhD thesis, Universität Hamburg, Von-Melle-Park 3, 20146 Hamburg, 2013.
- [Lud14] Prof. Thomas Ludwig. Hochleistungsrechnen. Vorlesung, 2014.

Quellen aktualisieren (ich hasse bibtex pflegen)