

# **Vorhersage von E/A-Leistung im Hochleistungsrechnen unter der Verwendung von neuronalen Netzen**

**— Bachelorarbeit —**

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

|                 |                                   |
|-----------------|-----------------------------------|
| Vorgelegt von:  | Jan Fabian Schmid                 |
| E-Mail-Adresse: | 2schmid@informatik.uni-hamburg.de |
| Matrikelnummer: | 6440383                           |
| Studiengang:    | Computing in Science - SP. Physik |
| Erstgutachter:  | Dr. Julian Kunkel                 |
| Zweitgutachter: | Prof. Dr. Thomas Ludwig           |
| Betreuer:       | Dr. Julian Kunkel                 |

Hamburg, den 17.12.2015

# Abstract

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Einleitung</b>  | <b>5</b>  |
| 1.1      | Motivation . . . . .   | 5         |
| 1.2      | Problemstellung . . . . .  | 5         |
| 1.3      | Ziele der Thesis . . . . .   | 6         |
| 1.4      | Strukturierung . . . . .   | 6         |
| <b>2</b> | <b>Hintergrund</b>   | <b>7</b>  |
| 2.1      | Dateisystem . . . . .  | 7         |
| 2.2      | HPC . . . . .  | 7         |
| 2.3      | Mistral . . . . .  | 7         |
| 2.4      | SIOX . . . . .   | 7         |
| 2.5      | Neuronale Netze . . . . .  | 7         |
| <b>3</b> | <b>Verwandte Arbeiten</b>  | <b>8</b>  |
| 3.1      | Kategorisierung (woanders hin?) . . . . .                              | 8         |
| 3.2      | Modellierungssansatz gegenüber Interpolationsansatz . . . . .          | 8         |
| 3.3      | Themenverwandtes . . . . .   | 9         |
| 3.4      | Fourier-Assisted Machine Learning . . . . .                            | 9         |
| 3.5      | Predicting Performance of Non-Contiguous I/O with Machine Learning . . | 10        |
| <b>4</b> | <b>Design &amp; Implementierung</b>                                    | <b>11</b> |
| <b>5</b> | <b>Evaluierung</b>   | <b>12</b> |
| <b>6</b> | <b>Anwendungsfälle</b>   | <b>13</b> |
| <b>7</b> | <b>Fazit</b>   | <b>14</b> |
|          | <b>Bibliography</b>  | <b>15</b> |
|          | <b>List of Figures</b>   | <b>16</b> |
|          | <b>List of Tables</b>  | <b>17</b> |
|          | <b>Listingverzeichnis</b>  | <b>18</b> |
|          | <b>Anhänge</b>   | <b>19</b> |



# 1. Einleitung

*Im folgenden wird zunächst kurz dargelegt mit welcher Problemstellung sich diese Thesis befasst, welches Ziel verfolgt wird, und wie dieser Text im Weiteren aufgebaut sein wird.*

## 1.1. Motivation

Hochleistungsrechnen ist in der Wissenschaft ein Thema mit zunehmender Signifikanz, viele komplexere Fragestellungen, insbesondere in den Naturwissenschaften und der Informatik, können z.B. nur in einer effizienten Weise durch Analyse einer Simulation eines Modells gelöst werden. Der extensive Rechenaufwand solcher Simulationen erfordert, dass Wissenschaftler nicht am eigenen Computer rechnen, sondern hierfür die Dienste eines Hochleistungszentrum in Anspruch nehmen. Die Entwicklung der Computer-Hardware in den vergangenen Jahrzehnten drängte die Hochleistungszentren dazu für den gewünschten Leistungsgewinn in stark parallelisierte Systeme zu investieren. Sodass, statt einzelner sehr schneller Prozessoren heutzutage viele Tausend Prozessoren vernetzt arbeiten. Diese horizontale Leistungssteigerung am Hochleistungsrechner umgeht die technischen Flaschenhälse, welche die Leistung eines einzelnen Prozessors beschränkt, die zur Verfügung stehende Leistung wird dadurch allerdings schwieriger nutzbar. Einerseits liegt dies am großen technischen Aufwand, der zur Vernetzung der Recheneinheiten notwendig ist, andererseits liegt es an der komplexen Programmierung der Software, welche die Parallelität des Rechners berücksichtigt. Um die Wissenschaftler beim Programmieren zu unterstützen, gibt es hilfreiche Tools zur Fehlerdiagnostik, Leistungsanalyse, Visualisierung des Programms und der Ergebnisse, sowie zum Parallelisieren von Programmcode.

## 1.2. Problemstellung

Im Bereich der Leistungsanalyse stellt sich unter anderem die Problematik der effizienten Ausnutzung der verschiedenen Puffer-Speicher (Caches), wie Arbeitsspeicher, und die direkt auf dem Prozessor liegenden Caches. Dabei stellt sich die Problematik des Vorhersagen der Dauer von Dateizugriffen (warum?). Diese Arbeit versucht diese Problematik mit dem Hilfsmittel neuronaler Netze zu lösen.

## 1.3. Ziele der Thesis

Main goal:

A neural network that is reliable in predicting performance of HPC-IO with sufficient quality

Subgoals:

Knowing which data can be provided by SIOX

Identify interesting (data mining) features, that can be derived from the available data

Having an understanding of what kind of neural network is suitable for the task

A measure for adequate quality of an IO-performance predictor

Benchmark of different predictors using different approaches

Implementation of a predictor module for SIOX for online evaluation of application performances

## 1.4. Strukturierung

Nachdem in diesem Kapitel die Metaebene der Thesis behandelt wird, soll das zweite Kapitel alle nötigen Hintergrundinformationen zum Verstehen der Thematik und der hier angewandten Ansätze liefern. Das dritte Kapitel beschäftigt sich mit bereits vorhandenen Ansätzen und Arbeiten zur Problemstellung. Im vierten Kapitel werden die Konzepte, sowie die Implementierung, der verschiedenen Modelle beschrieben, die als potenzielle Lösungen des Problems entwickelt und getestet wurden. Die Evaluierung der verschiedenen Lösungsansätze wird daraufhin im fünften Kapitel vorgenommen. Im sechsten Kapitel wird untersucht, welche Lösungsansätze für welche Anwendungsfälle am besten geeignet sind, um dann mit diesen Erkenntnissen im siebten Kapitel ein Fazit der Arbeit zu ziehen.

## 2. Hintergrund

*test*

### 2.1. Dateisystem

Wie werden Dateien abgelegt, wie greift man darauf zu. Wie funktioniert Caching?

### 2.2. HPC

Wofür HPC? Was zeichnet HPC aus? Dateisystem im HPC Besondere herausforderungen

### 2.3. Mistral

Speziell die Hardware von Mistral, von hier kommen die Benchmarks

### 2.4. SIOX

Ziele Vorgehen Wofür diese Arbeit? [KZB15] The SIOX framework [13,14] aims to become a holistic approach covering the full cycle of monitoring, analysis, machine learning of the adequate settings and their automatic enactment.

### 2.5. Neuronale Netze

Data Mining kurz allg. zu lösende Problemklasse Vor- und Nachteile

## 3. Verwandte Arbeiten

*Um weniger Umschreiben zu müssen führe ich zunächst zwei Kategorien von Lösungsansätzen ein. Danach erwähne ich kurz einige Arbeiten, dessen Themen sich mit dem dieser Arbeit überschneiden, und danach gehe ich noch etwas detaillierter auf Veröffentlichungen mit hoher Ähnlichkeit und Relevanz ein.*

### 3.1. Kategorisierung (woanders hin?)

Das Problem, die Zugriffszeiten auf Festplatten vorherzusagen, kann im wesentlichen durch zwei verschiedene Ansätze gelöst werden.

Zum einen kann man versuchen ein Modell des Festplattensystems zu erstellen, indem Hardwaredetails, wie die Rotationsgeschwindigkeit der Platte, die Reaktionszeit und Geschwindigkeit des Lesekopfs, sowie das Zusammenspiel der Komponenten, bekannt sind oder entsprechende Parameter werden durch gezielte Untersuchung approximiert. (**stimmt das? Referenz?**) Mit diesem möglichst exakten Modell können dann Zugriffe simuliert und so deren Aufwand vorhergesagt werden.

Der zweite Ansatz ist auch der dieser Arbeit, es wird von dem eigentlichen Festplattensystem abstrahiert und stattdessen ein mathematisches Modell gesucht, das aus gemessenen Leistungswerten von Festplattenzugriffen die Werte von neuen Zugriffen ableitet. Ich unterscheide daher im Folgenden zwischen dem Modellierungsansatz (in der Fachliteratur etwa als *analytic device modeling* und *simulation modeling* bezeichnet), bei dem versucht wird das Festplattensystem nachzubilden und dem Interpolationsansatz, bei dem ein numerisches Modell entwickelt wird. Da hier ohne Wissen über die inneren Zustände des Systems modelliert wird, wird dieser Ansatz auch *black-box modeling* bezeichnet. (Quelle? Fourier assisted ml)

### 3.2. Modellierungsansatz gegenüber Interpolationsansatz

Die Nachteile von Modellen mit Modellierungsansatz liegen insbesondere daran, dass sie aufwendig zu konfigurieren sind "In fact, one of us (Oldfield) spent several months configuring DiskSim to model an existing device" [CMW<sup>+</sup>13] (S.1) und naturgemäß schnell veralten, da sie jeweils an spezielle Hardware angepasst sind. Der Vorteil dagegen ist, dass sie bei korrekter Konfiguration sehr präzise sind, wie z.B. hier gezeigt [RW94].



Der Interpolationsansatz ist in der Anwendung einfacher und flexibler, da es sich automatisch an das System anpasst. Dafür erwartet man aufgrund der fehlenden analytischen Einsicht ins System eine etwas schlechtere Präzision.

Für die Anwendung im HPC-Bereich spielt der analytische Ansatz eine untergeordnete Rolle, da hier unterschiedliche Festplattensysteme zusammenarbeiten und stark mit der Netzwerkarchitektur verstrickt sind, sodass eine entsprechende Analyse des Systems zu aufwendig wird. "Furthermore, [...] building an accurate model or simulator using white box method cannot be a general solution in serving a variety of very different workloads" [ZLZ<sup>+</sup>10] (S.2, Zeile 20-24).

### 3.3. Themenverwandtes

Im HPC Bereich ist die Leistungsanalyse generell ein wichtiger Punkt, so wird beispielsweise das Scheduling-Algorithmen vom Dateisystem simuliert [LFC<sup>+</sup>], hier wird DiskSim [BSSG08] zur Vorhersage der Festplattenzugriffszeiten genutzt, dabei nutzt DiskSim einen Modellierungsansatz (analytical simulation). Eine weitere Arbeit, in der ein Modellierungsansatz genutzt wird stammt von Lebrecht et al. [LDK09].

Bei Arbeiten, in denen ein Interpolationsansatz genutzt wird, werden verschiedene Data-Mining bzw. stochastischen Methoden angewandt, beispielsweise eine Kombination aus regression trees und support vector machines [DLZC12], bagging classification und regression trees [ZLZ<sup>+</sup>10]. Verschiedene statistische Methoden werden von Kelly et al. untersucht [KCGK04].

### 3.4. Fourier-Assisted Machine Learning

Adam Crume et al. [CMW<sup>+</sup>13] gehen davon aus, dass der entscheidende Faktor bei der Vorhersage von Zugriffszeiten in der Erkennung von periodischen Mustern liegt. Diese Annahme ist für eine einzelne Festplatte gerechtfertigt, da man die Zugriffszeit grob in zwei Teile aufteilen kann, zum einen die Bewegung des Lesekopfs auf die richtige Spur und die Bewegung des Kopfes entlang der Spur zum richtigen Punkt, auch wenn diese beiden Bewegungen in der Realität überlappen. Jede Festplattenspur hat entsprechend des Radius eine andere Periodendauer.

Durch eine Fourier Analyse finden sie die Hauptfrequenzen heraus und können diese dann nutzen, um mit einem neuronalen Netz Vorhersagen zu treffen. Eine Schwierigkeit in dieser Arbeit ist unter anderem, dass durch die große Anzahl Perioden nur ein kleiner Ausschnitt der Festplatte in seiner Gesamtheit, also alle möglichen Paare von Ausgangs- und Endspuren, untersucht werden kann. Die Priorität solcher Frequenzen für die exakte Leistungsvorhersage, ist im komplexen HPC-E/A-System zunächst einmal nicht zu erwarten, müsste allerdings untersucht werden.

### 3.5. Predicting Performance of Non-Contiguous I/O with Machine Learning

Direkt aus dem Bereich des Hochleistungsrechnens stammt die Arbeit von Zimmer und Betke [KZB15]. Hier wird versucht mit Hilfe von decision trees die performantesten Parameter für nicht zusammenhängende Zugriffe auf Dateien durch ROMIO, einer Implementierung von MPI-2 I/O, zu finden. Dabei sagen sie mit den decision trees die Performance für die verschiedenen Parameter auf den Daten voraus, sodass sie anhand dieser Vorhersagen die besten Parameter finden. (**habe ich das richtig verstanden?**) Mit dieser Methode haben sie es geschafft zufriedenstellende Ergebnisse zu erzielen. Die Simplizität von decision trees, beispielsweise können diese nur Entscheidungen als eine Aneinanderreihung von linearen Separationen des Werteraums treffen, lässt vermuten, dass mit Hilfe von komplexeren Data Mining - Methoden, wie neuronalen Netzen noch bessere Ergebnisse erzielt werden könnten. Hier findet sich ein potenzielles Anwendungsgebiet der Ergebnisse dieser Bachelorarbeit.

## 4. Design & Implementierung

*test*

## 5. Evaluierung

*test*

## 6. Anwendungsfälle

*test*

## 7. Fazit

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

# Bibliography

- [BSSG08] John S. Bucy, Jiri Schindler, Steven W. Schlosser, and Gregory R. Ganger. The disksim simulation environment version 4.0 reference manual, 2008.
- [CMW<sup>+</sup>13] Adam Crume, Carlos Maltzahn, Lee Ward, Thomas Kroeger, Matthew Curry, and Ron Oldfield. Fourier-assisted machine learning of hard disk drive access time models. In *Proceedings of the 8th Parallel Data Storage Workshop, PDSW '13*, pages 45–51, New York, NY, USA, 2013. ACM.
- [DLZC12] Chengjun Dai, Guiquan Liu, Lei Zhang, and Enhong Chen. Storage device performance prediction with hybrid regression models. In *Proceedings of the 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT '12*, pages 556–559, Washington, DC, USA, 2012. IEEE Computer Society.
- [KCGK04] Terence Kelly, Ira Cohen, Moises Goldszmidt, and Kimberly Keeton. Inducing models of black-box storage arrays. Technical report, 2004.
- [KZB15] Julian Kunkel, Michaela Zimmer, and Eugen Betke. Using Machine Learning to Predict the Performance of Non-Contiguous I/O, 07 2015.
- [LDK09] AS Lebrecht, NJ Dingle, and WJ Knottenbelt. A performance model of zoned disk drives with i/o request reordering. pages 97–106. IEEE COMPUTER SOC, 2009.
- [LFC<sup>+</sup>] Yonggang Liu, Renato Figueiredo, Dulcardo Clavijo, Yiqi Xu, and Ming Zhao. Towards simulation of parallel file system scheduling algorithms with.
- [RW94] Chris Ruemmler and John Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27:17–28, 1994.
- [ZLZ<sup>+</sup>10] Lei Zhang, Guiquan Liu, Xuechen Zhang, Song Jiang, and Enhong Chen. Storage device performance prediction with selective bagging classification and regression tree. In *Network and Parallel Computing, IFIP International Conference, NPC 2010, Zhengzhou, China, September 13-15, 2010. Proceedings*, pages 121–133, 2010.

## List of Figures



## List of Tables

# Listingverzeichnis

# Anhänge

## A. Anhangskapitel

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## Erklärung

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen, als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internetquellen – benutzt habe, die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

**Optional:** Ich bin mit der Einstellung der Bachelor-Arbeit in den Bestand der Bibliothek des Fachbereichs Informatik einverstanden.

Hamburg, den 01.01.2012 .....