

TECHNISCHE UNIVERSITÄT DRESDEN

ZENTRUM FÜR INFORMATIONSDIENSTE  
UND HOCHLEISTUNGSRECHNEN  
PROF. DR. WOLFGANG E. NAGEL

## Analyse eines Forschungsthemas

### Vergleich existierender Tools zur Messung und Analyse der Performance von HPC Clustern

Christian Kroh  
(Mat.-Nr.: 3755154)

Hochschullehrer: Prof. Dr. Wolfgang E. Nagel  
Betreuer: Dr. Holger Brunst

Dresden, 31.03.2018

---

# Declaration of Authorship

I hereby certify that this report on the topic of

*Vergleich existierender Tools zur Messung und Analyse der Performance von HPC Clustern*

which I presented today to the examination office of the Faculty of Computer Science at the Technische Universität Dresden has been composed by me and is my own work, unless stated otherwise. No other person's work has been used without due acknowledgement in this report. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.

Dresden, the 31.03.2018

Christian Kroh

---

## **Kurzfassung**

Das regelmäßige Überprüfen von High Performance Computing (HPC) Cluster bezüglich Performance-Änderungen von einzelnen Recheneinheiten oder des gesamten Systems ist notwendig um Probleme frühzeitig zu erkennen und zu beheben. Solche können durch fehlerhafte Konfiguration, falsch installierte Software oder defekte Hardware während des Betriebs durch Updates oder Ermüdung auftreten.

Die geminderte Leistungsfähigkeit einzelner Knoten könnte jedoch im Fall von hoch-parallelen Anwendungen mit einigen sequentiellen Abschnitten zu einer signifikanten Laufzeiterhöhung führen, da der auf den schwachen Knoten ausgeführte Anwendungsteil die Ausführung auf den restlichen Knoten verzögert. Ohne eine aufwendige Untersuchung der Anwendungsperformance könnte es sein, dass eine solche nachteilige Ausführung unerkannt bleibt.

Die Lösung für dieses Problem könnte ein System sein, dass das HPC Cluster regelmäßigen Tests unterzieht und deren Ergebnisse analysiert ohne die Verfügbarkeit des Clusters erheblich einzuschränken. Entsprechende Tests können Anwendungen sein, die bestimmte Hardwarekomponenten überprüfen oder die Leistungsfähigkeit des Clusters in speziellen Anwendungsszenarien dokumentieren. Eine andere Form der Performance-Analyse ist das permanente Auslesen und Sammeln von Systeminformationen, was als Monitoring bezeichnet wird.

Mithilfe eines solchen Systems könnten Systemadministratoren schnell erkennen, ob einzelne Knoten im Vergleich zu Knoten mit identischer Hardware, weniger Rechenleistung zur Verfügung stellen. Auch könnten die Veränderungen durch ein Software-Update Einfluss auf die Performance des Clusters in speziellen Anwendungsszenarien haben und so festgestellt werden. Dies würde es ermöglichen frühzeitig defekte Knoten zu isolieren oder Empfehlungen bezüglich der Verwendung von Softwarepaketen auszusprechen.

In meiner Arbeit werde ich bisherige Implementierungen und Praktiken aus den Bereichen der Anwendungsperformance-Analyse, dem System-Monitoring und der Performance-Analyse einzelner Knoten vorstellen und diese anhand der Eigenschaften ihrer Funktionalität, Anwendbarkeit, Nutzerfreundlichkeit und Erweiterbarkeit miteinander vergleichen.

## **Abstract**

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Tools</b>	<b>3</b>
2.1	Monitoring Tools . . . . .	3
2.1.1	Dataheap . . . . .	3
2.1.2	Ganglia . . . . .	3
2.1.3	PerfMiner . . . . .	4
2.1.4	SuperMon . . . . .	4
2.1.5	Lightweight Distributed Metric Service . . . . .	4
2.1.6	ClusterProbe . . . . .	4
2.1.7	Disparity . . . . .	4
2.1.8	NWPerf . . . . .	4
2.2	Experiment-basierte Performance Analyse Tools . . . . .	4
2.3	Application Performance Analyse Tools . . . . .	4
2.3.1	Tracing . . . . .	4
2.3.2	Profiling . . . . .	4
2.3.3	Visualization . . . . .	4
<b>3</b>	<b>Anwendungstest</b>	<b>5</b>
3.1	Monitoring: Dataheap . . . . .	5
3.2	Experiment: ? . . . . .	5
3.3	Application Performance: Score-P . . . . .	5
<b>4</b>	<b>Aktuelle Entwicklung</b>	<b>6</b>
<b>5</b>	<b>Fazit</b>	<b>7</b>

# 1 Einleitung

Der Betrieb von High-Performance Computing Systemen ist durch den hohen Energiebedarf

Wieso Cluster Performance ermitteln? -> Vergleich ... Benchmarking, Top500

Beschreibe HPC Cluster Ausfälle ...[?]

Auswirkung von schwachen Knoten auf over-all performance von parallelen anwendungen

Vergleiche Tools [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?]

Kategorien Monitoring, Experiment Analyse, Application Performance in HPC Performance Analyse

Beispiele

**Monitoring** Ganglia, Dataheap, PerfMiner, SuperMon, Lightweight Distributed Metric Service, ClusterProbe, Disparity, NWPerf

**Experiment Analyse** PHM (Performance Health Monitor), ZENTURIO, PerfExplorer, PerfBase

**Application Performance** Score-P, PerfExpert, HPCToolkit, Open!SpeedShop, PerfSuite, Perf (ZIH-Projekt), Valgrind-based Tools, TAU

## 2 Tools

Beschreibe existierende Tools und ordne diese den Kategorien Monitoring, Experiment-basiert und Application Performance zu. In diesen Bereichen zeige Eigenheiten der Tools bzgl. Möglichkeiten der Performance Messung und Analyse auf. Verdeutliche wie, wo und von wie vielen Nutzern diese Tools aktuell verwendet werden.

### 2.1 Monitoring Tools

#### 2.1.1 Dataheap

2012

#### 2.1.2 Ganglia

Von der Universität Berkeley von Kalifornien zusammen mit Intel im Jahr 2004 eingeführtes Tool zur parallelen Datenerfassung auf verteilten Systemen.

Durch den einfachen Aufbau und der daraus resultierende schnellen Installation, ist Ganglia zu einem beliebten Monitoring-Werkzeug geworden. Mithilfe seiner erweiterbaren Metriken, kann es an jede beliebige Anwendung angepasst werden und ist dafür ausgelegt auf Systemen mit bis zu 2000 Knoten zu skalieren.

Trotz des relativen Alters der Software, wird das Tool noch immer auf tausenden HPC Systemen verwendet [?]. Ein limitierender Faktor von Ganglia ist allerdings seine maximale Update-Frequenz von höchstens einer Sekunde, was in Szenarien die eine größere zeitliche Auflösung benötigen, nicht mehr ausreicht.

**2.1.3 PerfMiner**

**2.1.4 SuperMon**

**2.1.5 Lightweight Distributed Metric Service**

**2.1.6 ClusterProbe**

**2.1.7 Disparity**

**2.1.8 NWPerf**

**2.2 Experiment-basierte Performance Analyse Tools**

**2.3 Application Performance Analyse Tools**

**2.3.1 Tracing**

**2.3.2 Profiling**

**2.3.3 Visualization**

## **3 Anwendungstest**

Teste eine Auswahl von Tools selber aus und beschreibe Erfahrungen.

### **3.1 Monitoring: Dataheap**

### **3.2 Experiment: ?**

### **3.3 Application Performance: Score-P**



## **4 Aktuelle Entwicklung**

Was sind die aktuell "besten" Tools in dem jeweiligen Bereich bzgl. ausgewählter Eigenschaften.

## **5 Fazit**

## Appendix