Ganzheitliches Verständnis von Nutzungs- und Systemverhalten als Zugang zu effizienterem Hochleistungsrechnen

Julian M. Kunkel

Universität Hamburg

5. Juni 2013

Hochleistungsrechnen – Motivation

Wissenschaftliche Anwendungen haben einen hohen Bedarf an

- Rechenzeit
- Arbeitspeicher
- Speicherkapazität

Ein „normaler” PC/Server ist meist nicht leistungsfähig genug

$\Rightarrow$ Parallele Nutzung vieler Prozessoren bzw. Server

Titan Supercomputer: Schnellster Rechner der Welt

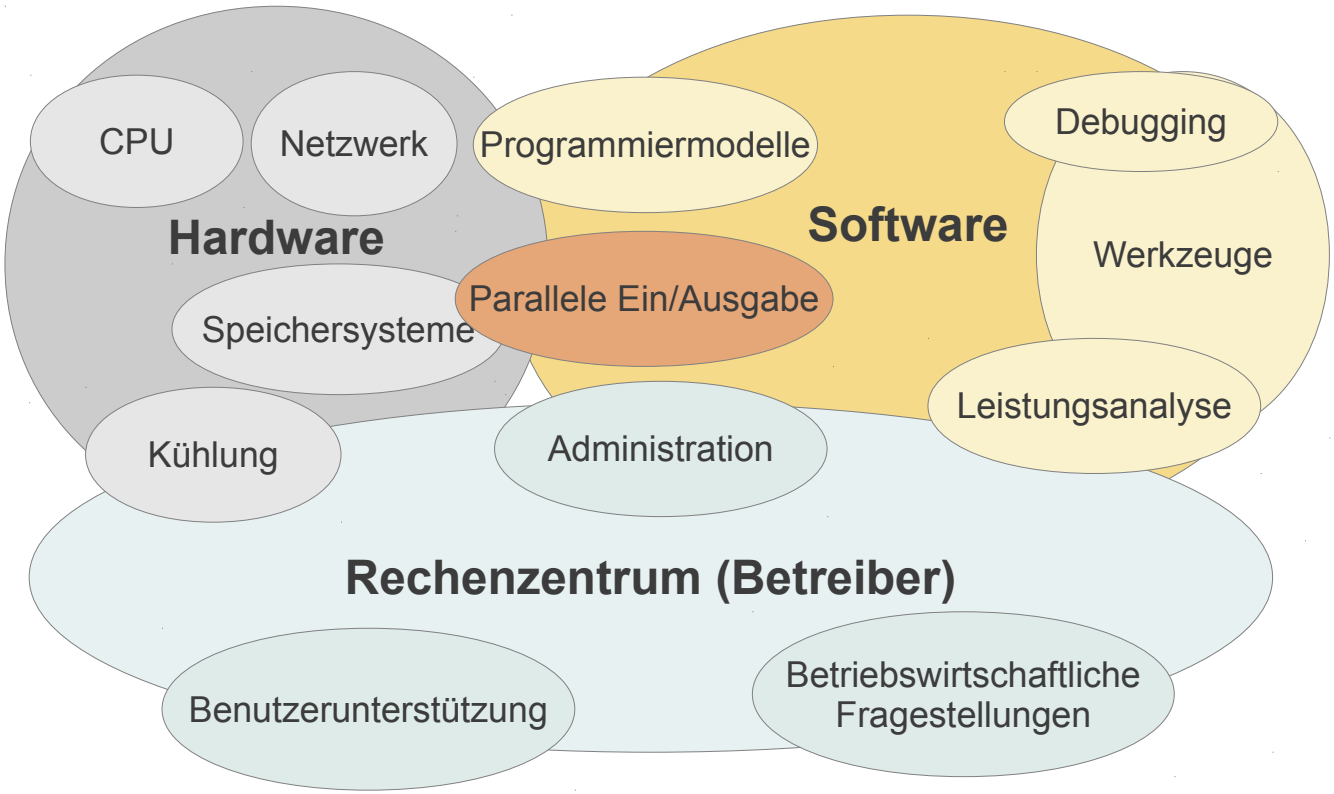
Titan – Frontansicht [*Quelle: Wikipedia, Titan_(supercomputer)*]

- 200 Racks auf 400 m^2
- 18.688 Rechenknoten:
  - 1 AMD Opteron 16-core CPU
  - 1 Nvidia Tesla GPU
  - 32 GByte RAM
- Max. Rechenleistung: 27 PFlop/s
- Hauptspeicher: 710 TByte
- Speicherkapazität: 40 PByte
- Preis: $ 100 Millionen
- Leistungsaufnahme: 8.2 MW

Definition: Hochleistungsrechnen

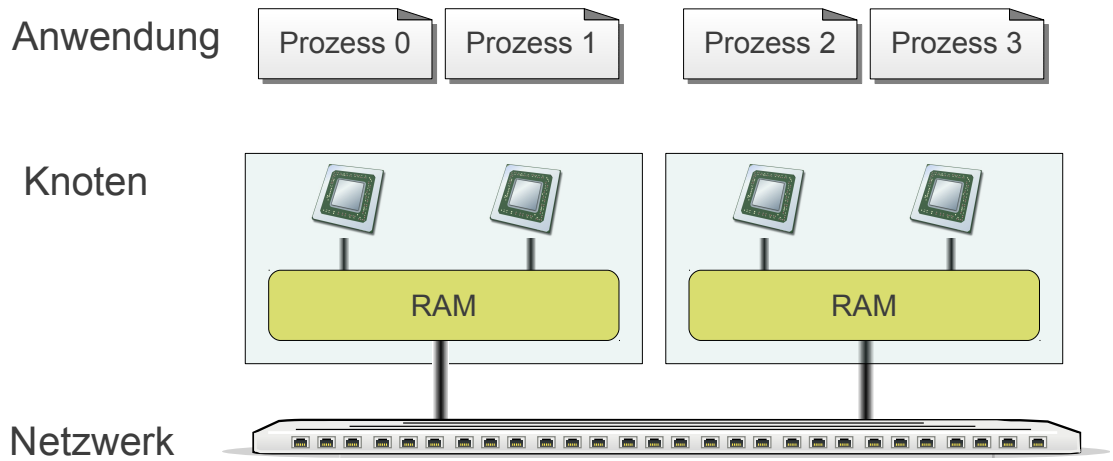
- Programmierung von Anwendungen mit hohem Ressourcenbedarf
- Disziplin der Herstellung und Nutzung von Supercomputern

Ausgewählte Teilgebiete



Parallele Programmierung

- Ziel: Viele CPUs kooperieren bei der Problemlösung
- Vorgehen: Parallele Bearbeitung von Daten oder Aufgaben
- Kooperation erfordert Koordination und Datenaustausch
- Programmiermodell definiert formale Spezifikation



Anwendung mit vier Prozessen auf zwei Knoten

Programmiermodell Nachrichtenaustausch

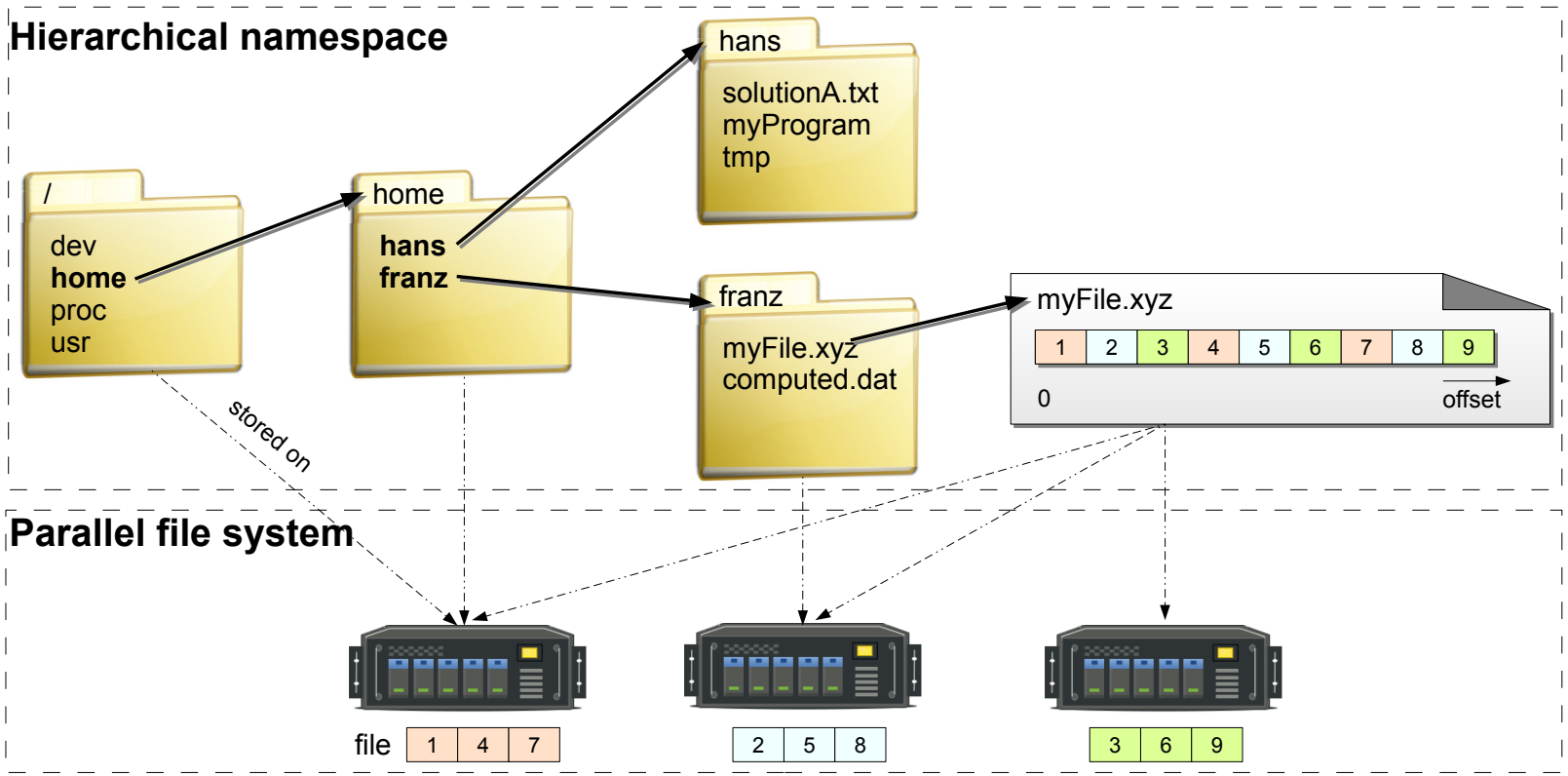
Message Passing Interface (MPI)

- Bibliothek für C und Fortran
- Expliziter Datenaustausch und Synchronisation
  - Punkt-zu-Punkt-Kommunikation
    - `MPI_Send()`, `MPI_Recv()`, ...
  - Kollektive Operationen
    - `MPI_Bcast()`, `MPI_Scatter()`, ...
  - Parallele Ein-/Ausgabe

```
1 if (process == 0){
2   int res = taskA();
3   MPI_send(& res, 1, MPI_INT, 1, ...);
4 }else if (process == 1){
5   int remote_res;
6   int res = taskB();
7   MPI_recv(& remote_res, 1, MPI_INT, 0, ...);
8   taskC(res, remote_res);
9 }
```

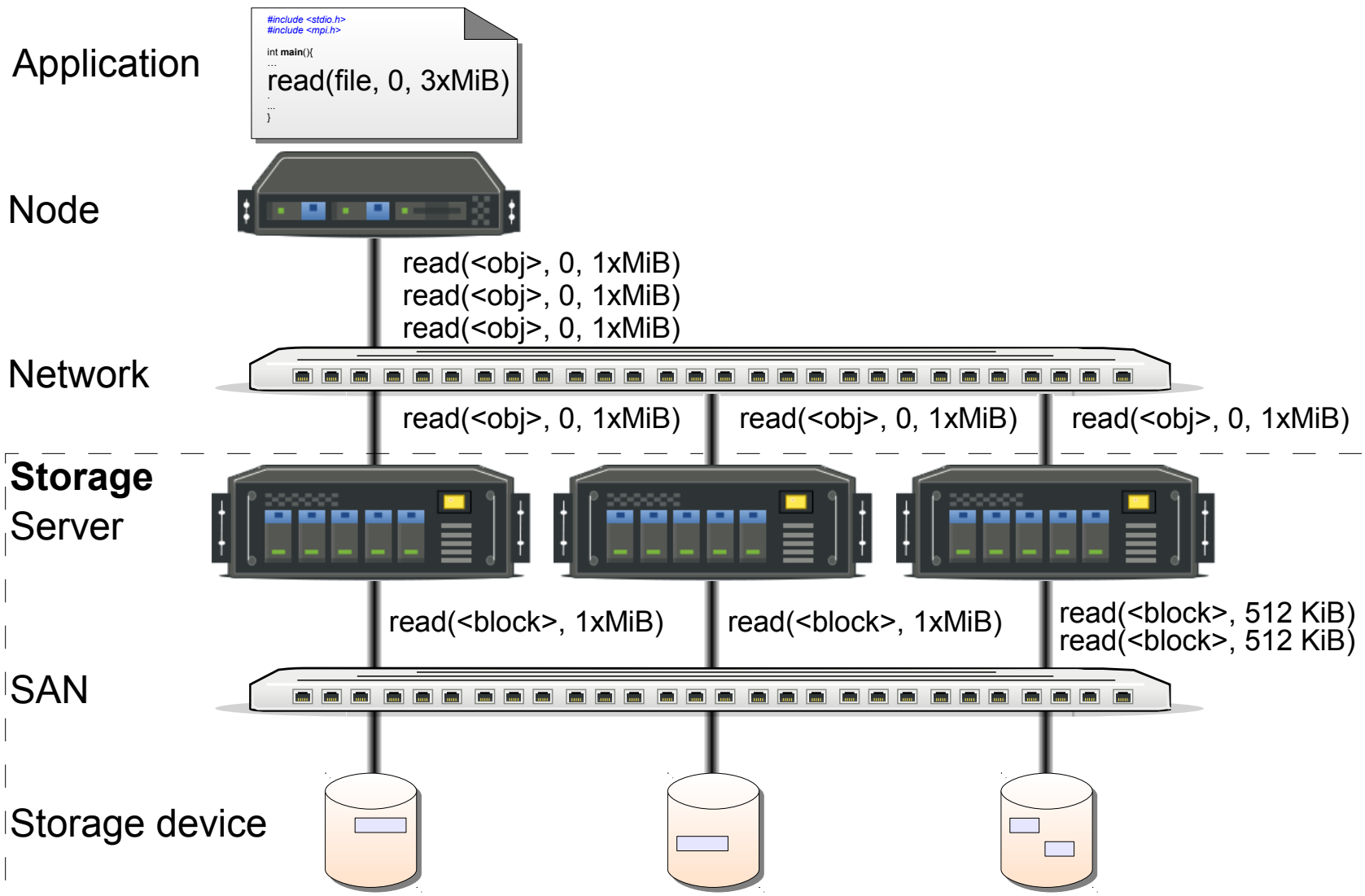
Parallele Dateisysteme

- Verteilen Daten eines Namensraums über viele Server
- Ziel: Aggregierte Leistungsfähigkeit aller Server
- Parallel: Gleichzeitiger Zugriff mehrerer Prozesse auf eine Datei



Exemplarischer hierarchischer Namensraum

Ausgeführte Operationen beim Dateizugriff



Ausgeführte Operationen um 3 MiB Daten einer Datei zu lesen

Leistungsanalyse und Optimierung

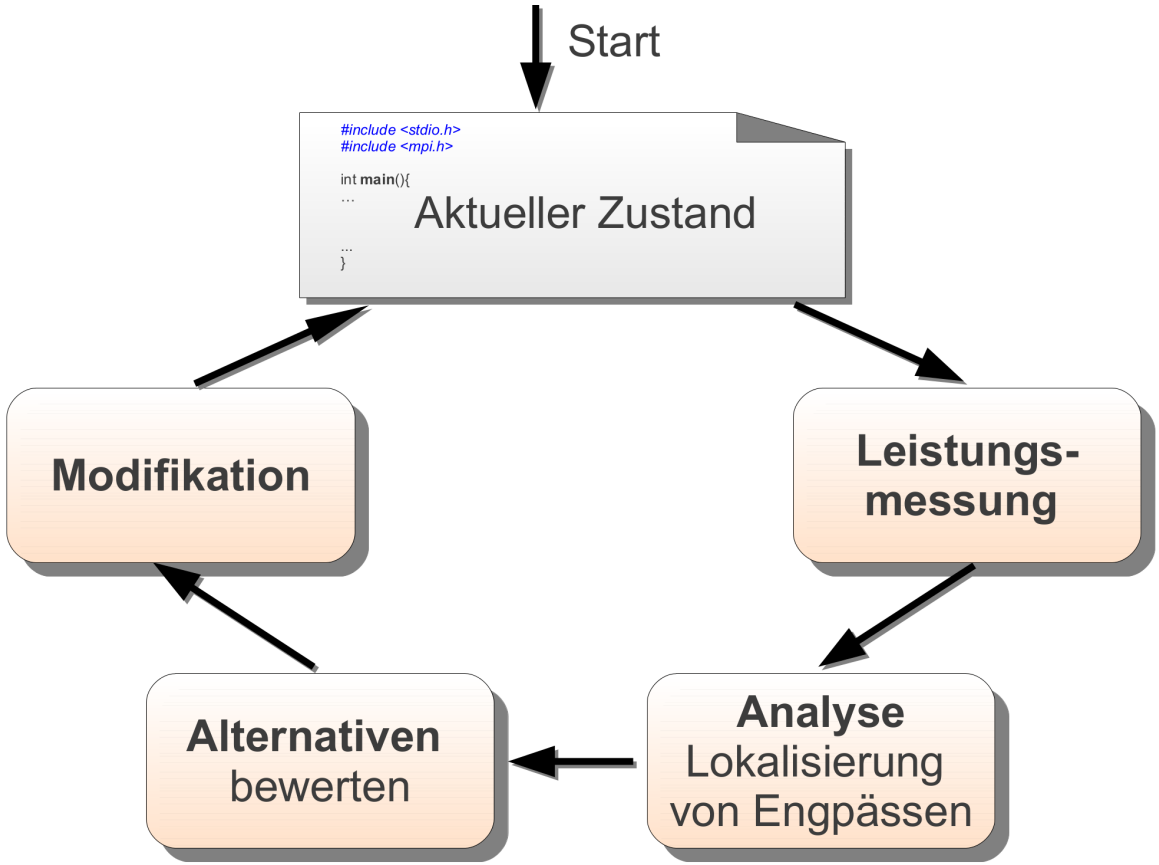
Motivation

Anwendungen nutzen oft nur wenig vorhandener Rechenleistung

Ursachen

- Algorithmus/Code passt nicht zur Hardware
- Ineffiziente Berechnung, Kommunikation oder E/A

Vorgehensweise bei der Leistungsoptimierung



Traditionelles Vorgehen – Iterative Optimierung

Leistungsanalyse

Fragestellungen

1. Wie analysieren wir das Laufzeitverhalten von Anwendungen?
   - Spurdaten-Werkzeuge erfassen Aktivitäten
   - Manuelle Analyse des beobachteten Verhaltens
2. Wie bewerten wir beobachtetes Verhalten?
   - Relation von Leistungsfähigkeit und Beobachtung
   - Simulation vereinfacht theoretische Betrachtungen
3. Wie können wir die Leistungsfähigkeit von Hardware erfassen?
   - Benchmarks erzeugen definierte Lasten
   - Herstellerangaben als Referenz
4. Welche Optimierungen sollten am System durchgeführt werden?
   - Zielgerichtet durch Nutzungsstatistiken und Benutzerdialog

Forschungsarbeiten

Persönliches Ziel

Ganzheitliches Verständnis (und Optimierung) von Supercomputern und ihrer Nutzung

Forschungsschwerpunkte

- Leistungsanalyse und Bewertung
- Parallele Ein-/Ausgabe

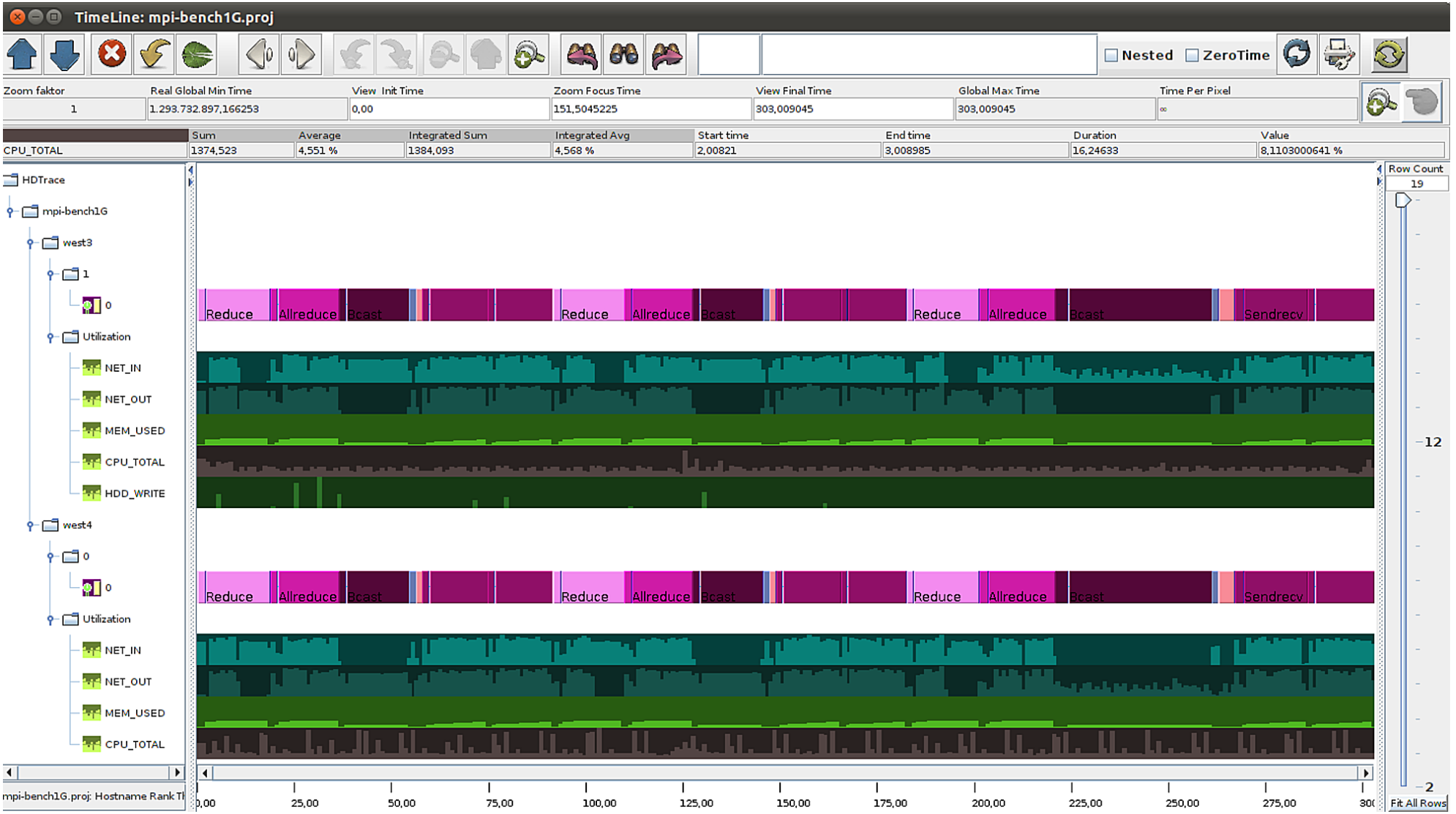
Weitere Forschungsthemen

- Wissenschaftliche Softwareentwicklung
- Erfassung von Systemnutzung
- Betriebswirtschaftliche Fragestellungen

Spurdaten-Umgebung HDTrace

- Werkzeug für die Erfassung und Analyse von Spurdaten
- Betrachtung von Anwendungs- und Systemverhalten
- Forschungsvehikel für Analysemöglichkeiten
  - Statistiken von Betriebsystem und Dateisystem
  - Energieverbrauch von Anwendungen
  - Aktivitäten von parallelen Dateisystemen
  - MPI-Interna (Kollektive Operationen, Datentypen und E/A-Zugriffe)
  - Alternative Visualisierungsmöglichkeiten

Sunshot

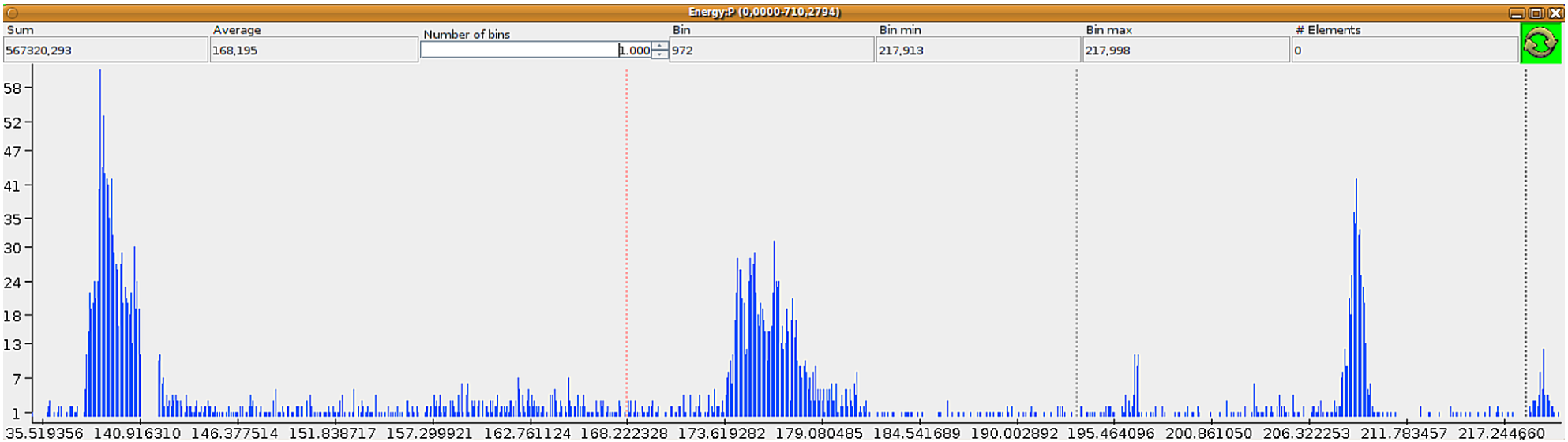


Sunshot Timelines und Statistiken

Energieverbrauch

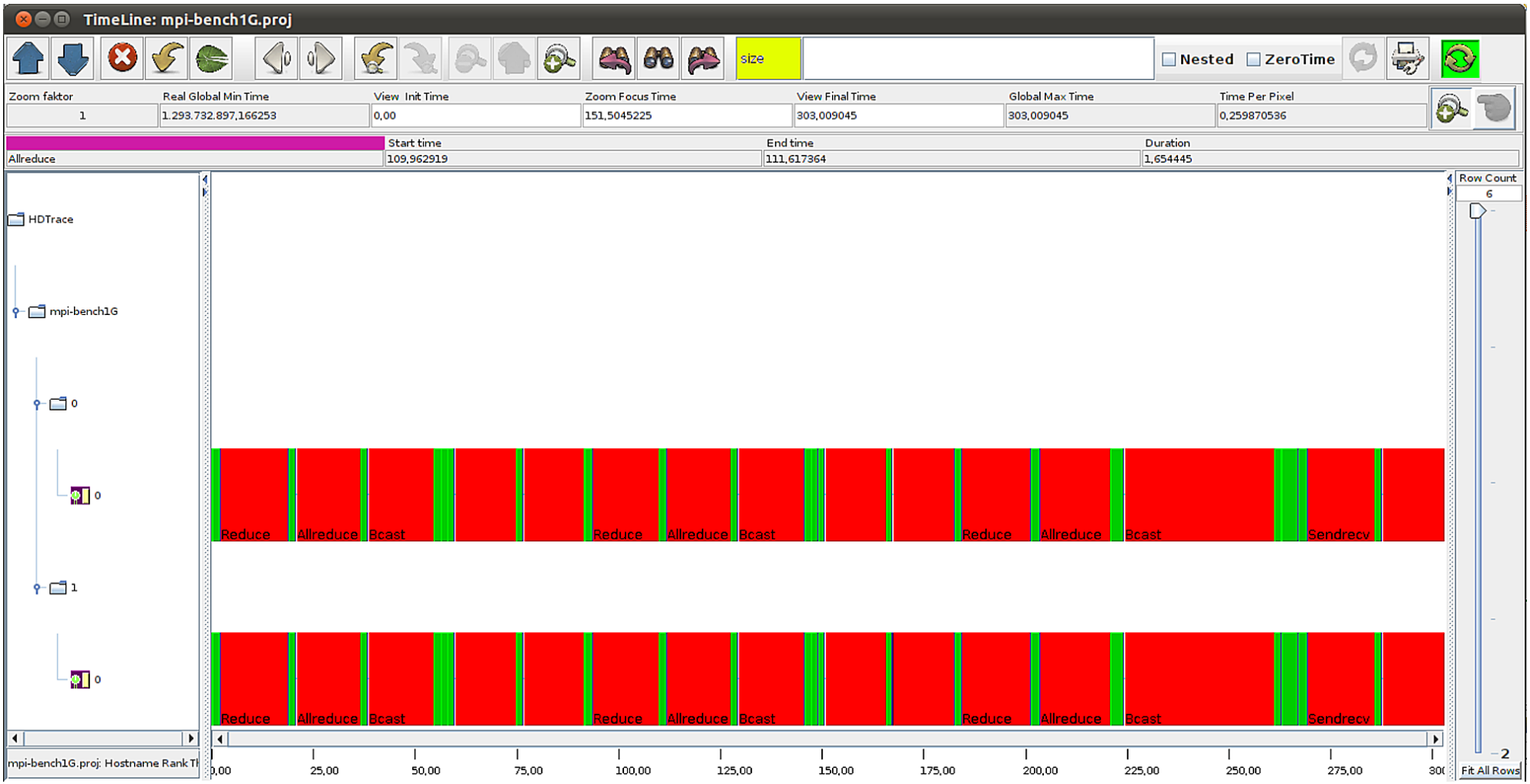
Energiemetriken von einem HPCC-Lauf (globale Minima berücksichtigt)

Histogramme



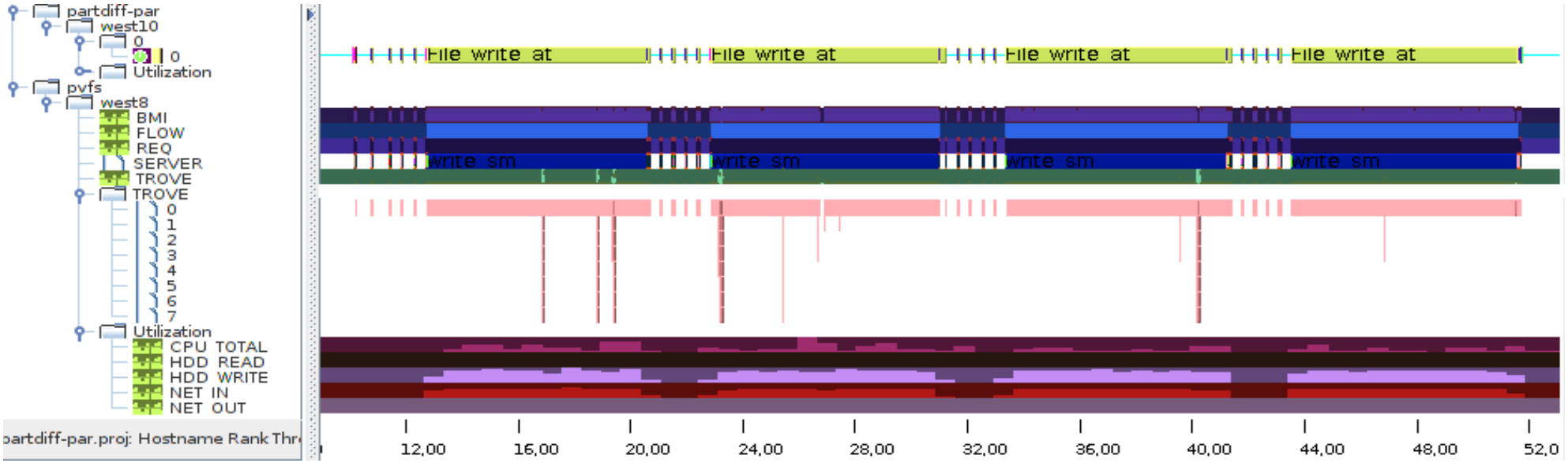
Histogram der Leistungsaufnahme für den HPCC-Lauf (in Watt)

Heat-Maps



Farbliche Markierung nach der Größe eines Zugriffs (hier zwei)

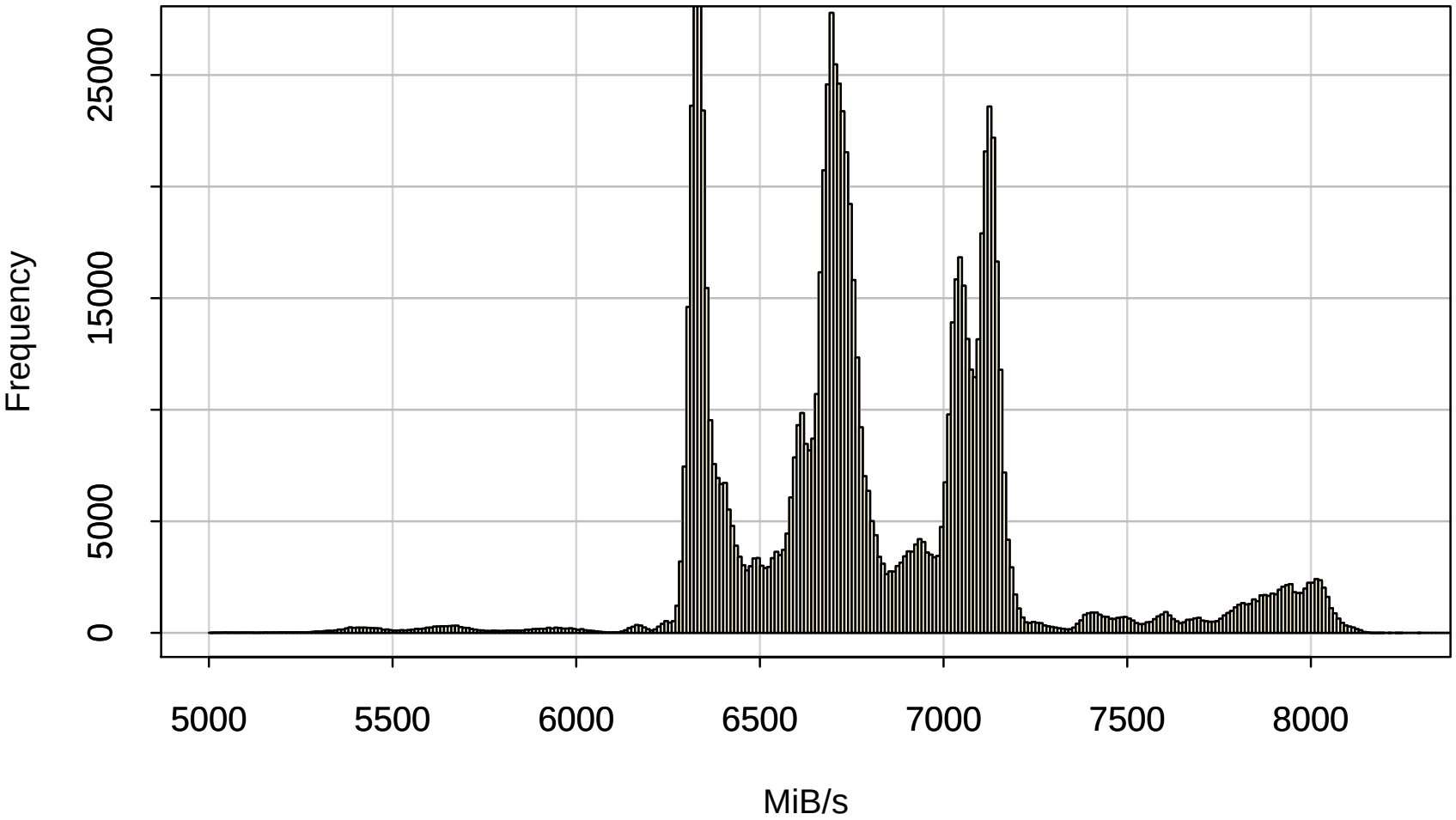
Instrumentierung von PVFS



Visualisierung von PVFS Client- und Server-Aktivitäten

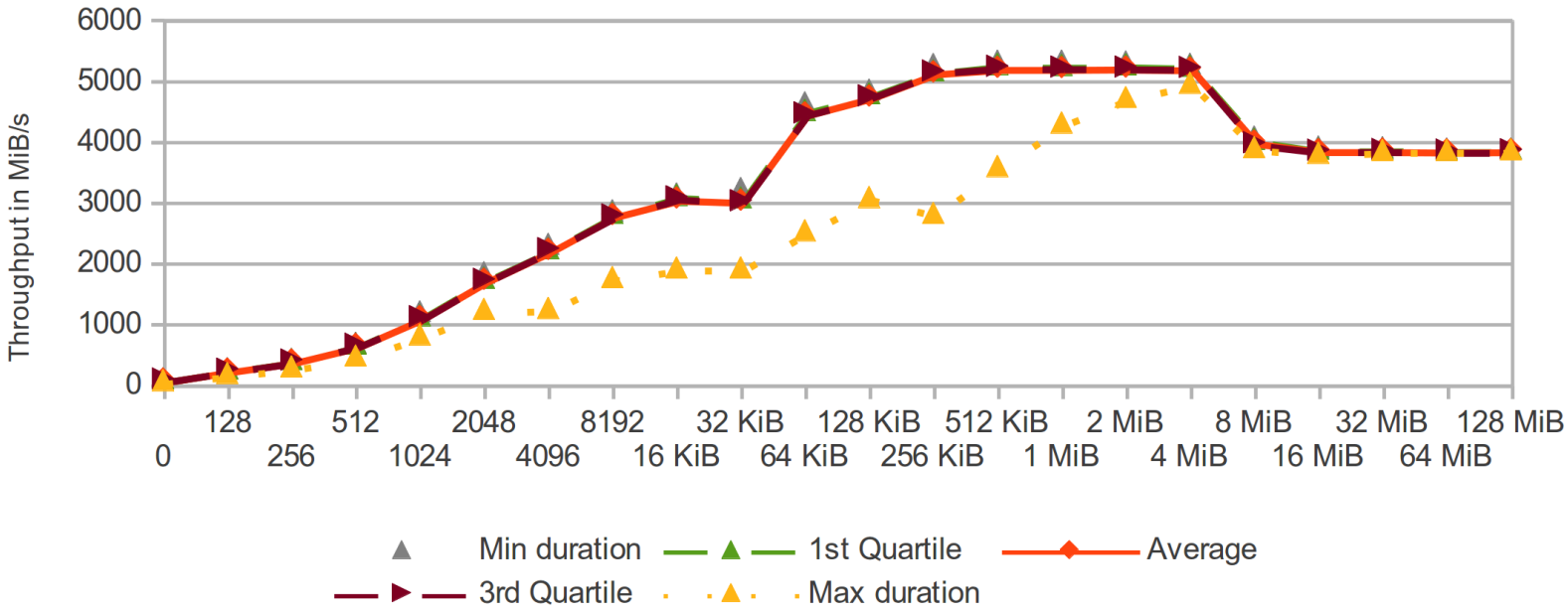
Beispiele für die Komplexität der Leistungsbewertung aufgrund von Hardware- und Softwareverhalten

Arbeitsspeicher



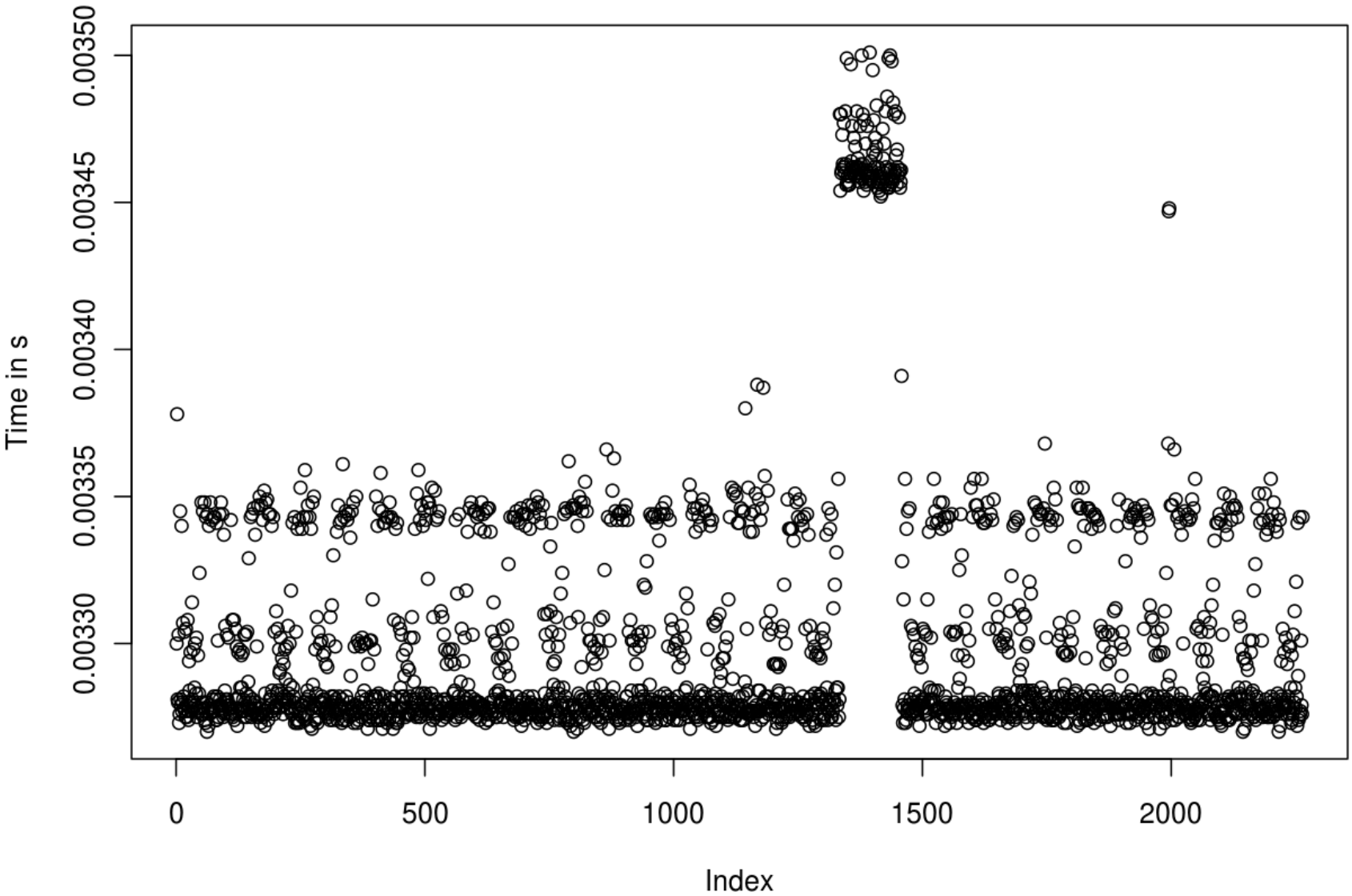
Histogramm für 1 MiB Schreibzugriffe auf 1 GByte Speicher, 1000 Wiederholungen

MPI Punkt-zu-Punkt-Kommunikation



Durchsatz bei lokaler Kommunikation zwischen zwei Sockel

MPI Punkt-zu-Punkt-Kommunikation



Intra-Sockel Einzelmessungen für 8 MiB Nachrichten

Leistungsbewertung

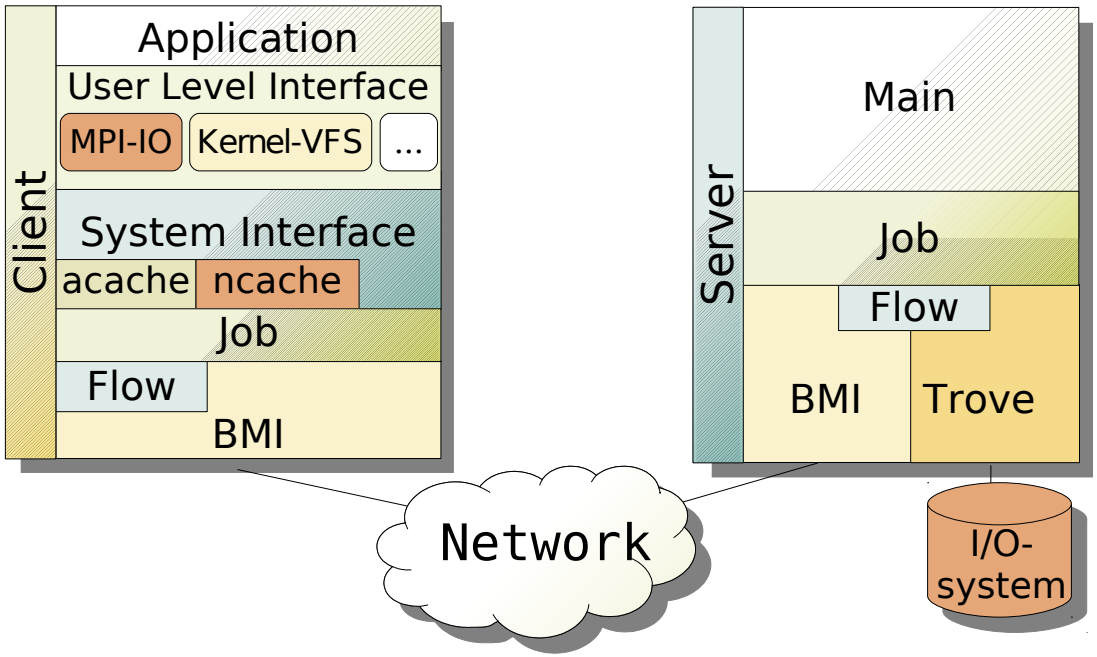
Erklärung von gemessener Leistung ist nicht trivial

Einflussfaktoren

- Hardware
- Softwarekomplexität
  - Betriebsystemeinflüsse
  - Beteiligung mehrerer Schichten
- Interaktionen von Optimierungen
- Gemeinsame Nutzung von Ressourcen
- Größe des Systems und lange Programmlaufzeiten

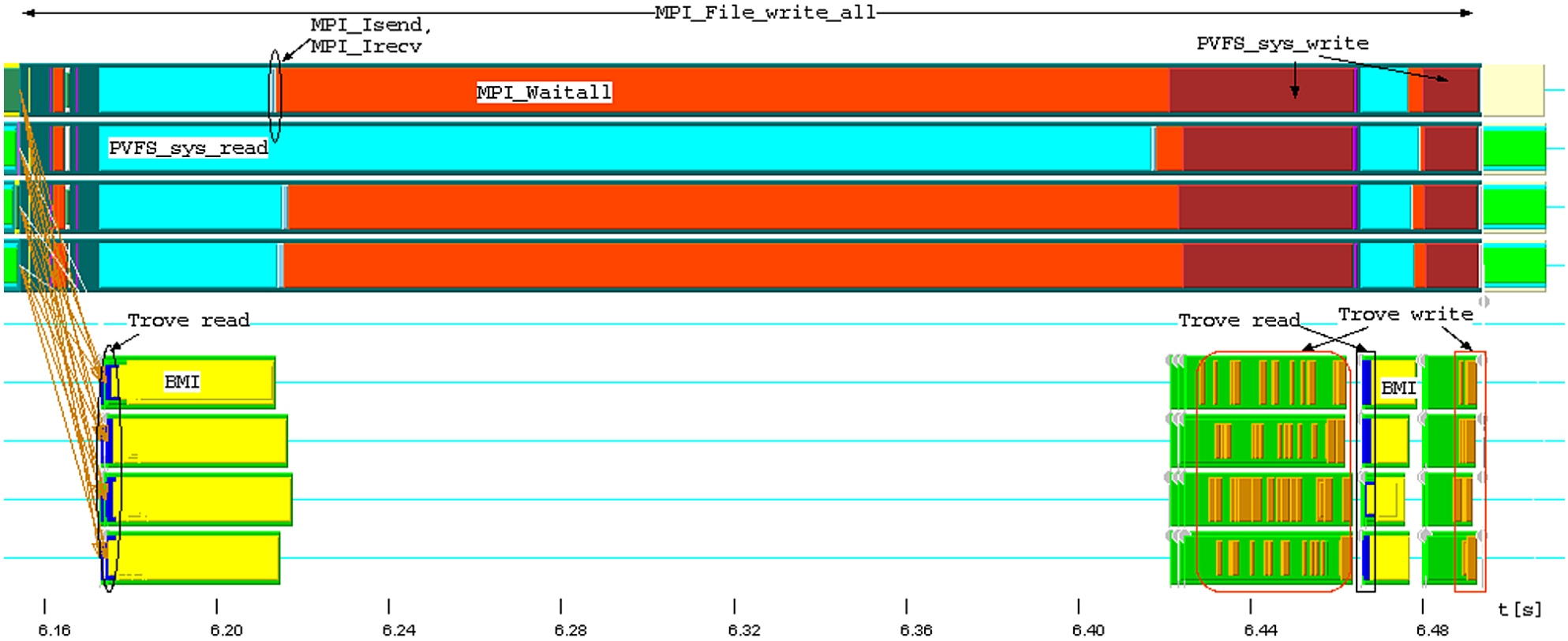
Modifikation des parallelen Dateisystems PVFS

- Leistungsanalysemodul als Ersatz für die Persistenz-Schicht
- Optimierungen von kleinen Dateien
  - DKRZ: 160M Dateien, 30% aller Dateien $<$ 8 KiB, 90% $<$ 2 MiB
- Instrumentierung zur Spurdatenanalyse



PVFS Software-Architektur

Leistungsanalyse von PVFS mittels Spurdaten



Client-Aktivität und ausgelöste Server-Aktionen

Programmierbarer E/A-Benchmark – Parabench

Ziel: Beliebige (Metadaten) Lasten evaluieren

Funktionen

- Parallele Interpretation und Ausführung einer Last-Spezifikation
- Einheitliche Ergebnisausgabe
  - Diagrammerstellung (mit R)
  - Metriken: Zeit, Durchsatz, IO-Operationen/s
- Behandlung von E/A-Fehlern

Beispielhafte Last und Ausführung

Benchmarking von individuellen E/A-Operationen vgl. `fileop`

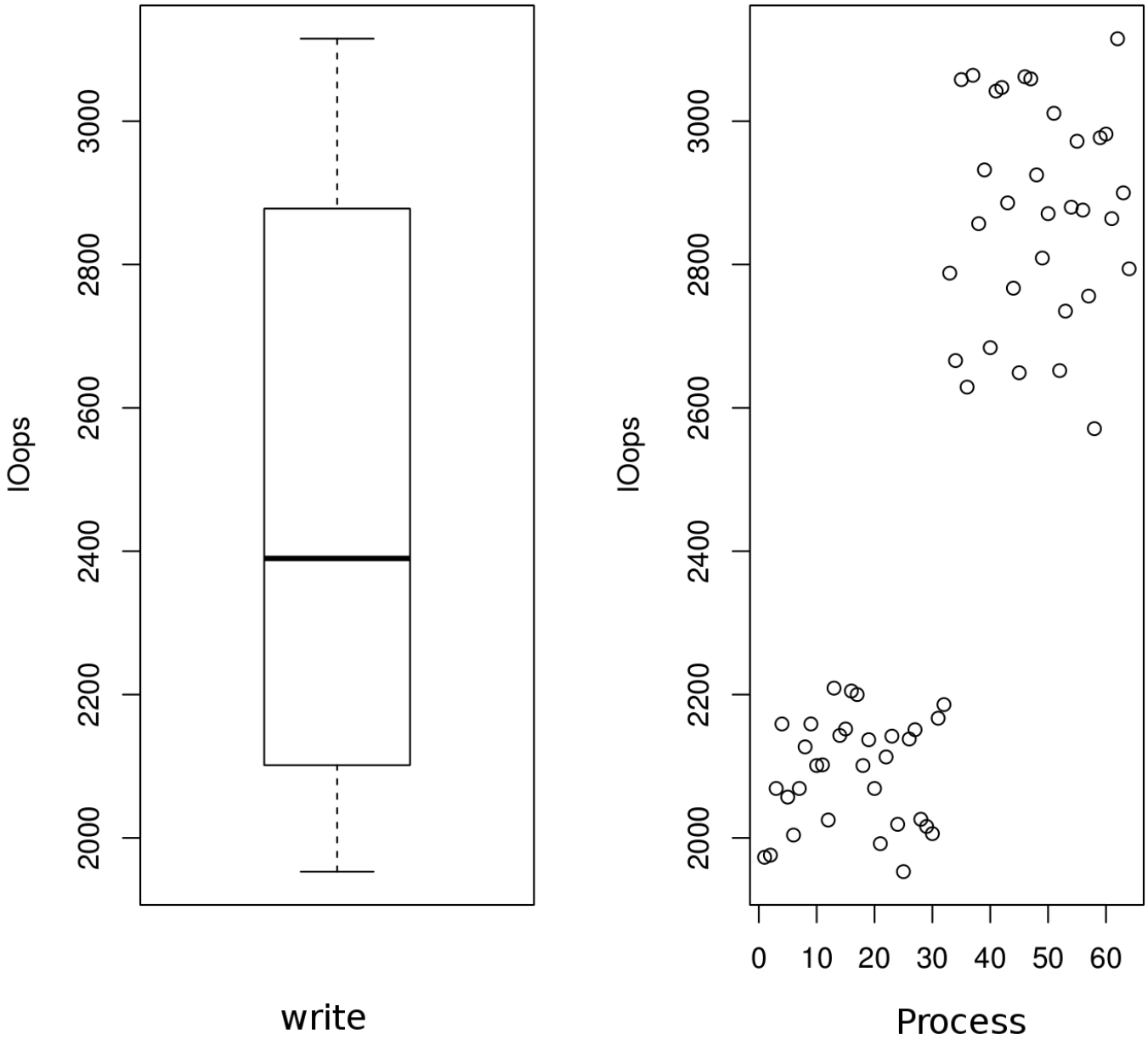
- Eigener Ordner pro Prozess und 100.000 Dateien
- 64 Prozesse auf zwei Knoten der Blizzard

```
********************* TreeTime Report *********************
 [P]  [#]  [+] [*] [event]                 [Walltime] [avgTime]  [minTime]  [maxTime]       [avgTP]        [minTP]        [maxTP]  [IOops/s]     [Volume]    [Calls]  [ErrorCalls]
all    1    0      create                5034.817078  0.000787   0.000119  30.104497             -              -              -       1271            -    6400000            0
all    2    0      write                   51.186087  0.000008   0.000069   2.863710  465.04 MiB/s     1.33 KiB/s     53.90 MiB/s     125033     23.25 GiB    6400000            0
all    3    0      write large             83.463072  0.000130   0.000369  15.090957    7.14 GiB/s    64.71 KiB/s      2.52 GiB/s       7668     596.05 GiB     640000            0
all    4    0      stat                  1155.611306  0.000181   0.000763   2.959044             -              -              -       5538            -    6400000            0
all    5    0      read                  1544.565378  0.000241   0.000639   9.058045    15.41 MiB/s    430.56   B/s      5.82 MiB/s       4143     23.25 GiB    6400000            0
all    6    0      read large             330.349486  0.000516   0.006747   1.486906     1.80 GiB/s    656.77 KiB/s    141.35 MiB/s       1937     596.05 GiB     640000            0
all    7    0      delete                5652.365149  0.000883   0.001327  17.340450             -              -              -       1132            -    6400000            0
all    8    0      delete large           441.955087  0.000691   0.001407  18.867001             -              -              -       1448            -     640000            0

[P]          - Process rank
[#]          - Execution order of the time command
[+]          - Parent in case of nested time command
[event]      - Time event label
[*]          - Starred columns have detailed times

****************** Command Report *********************
 Write         7040000 successful /            0 failed
 Read          7040000 successful /            0 failed
 Delete        7040000 successful /            0 failed
 Mkdir              65 successful /           63 failed
 Create        6400000 successful /            0 failed
 Stat          6400000 successful /            0 failed
```

Ausgabe von aggregierten Metriken für individuelle E/A-Operationen



Statistische Auswertung von Parabench für individuelle E/A-Operationen

BMBF-Projekt: SIOX

Motivation

- Mangel an Werkzeugen für die Bewertung von Anwendungs-E/A
- Vorhandene Optimierungen müssen manuell gesteuert werden

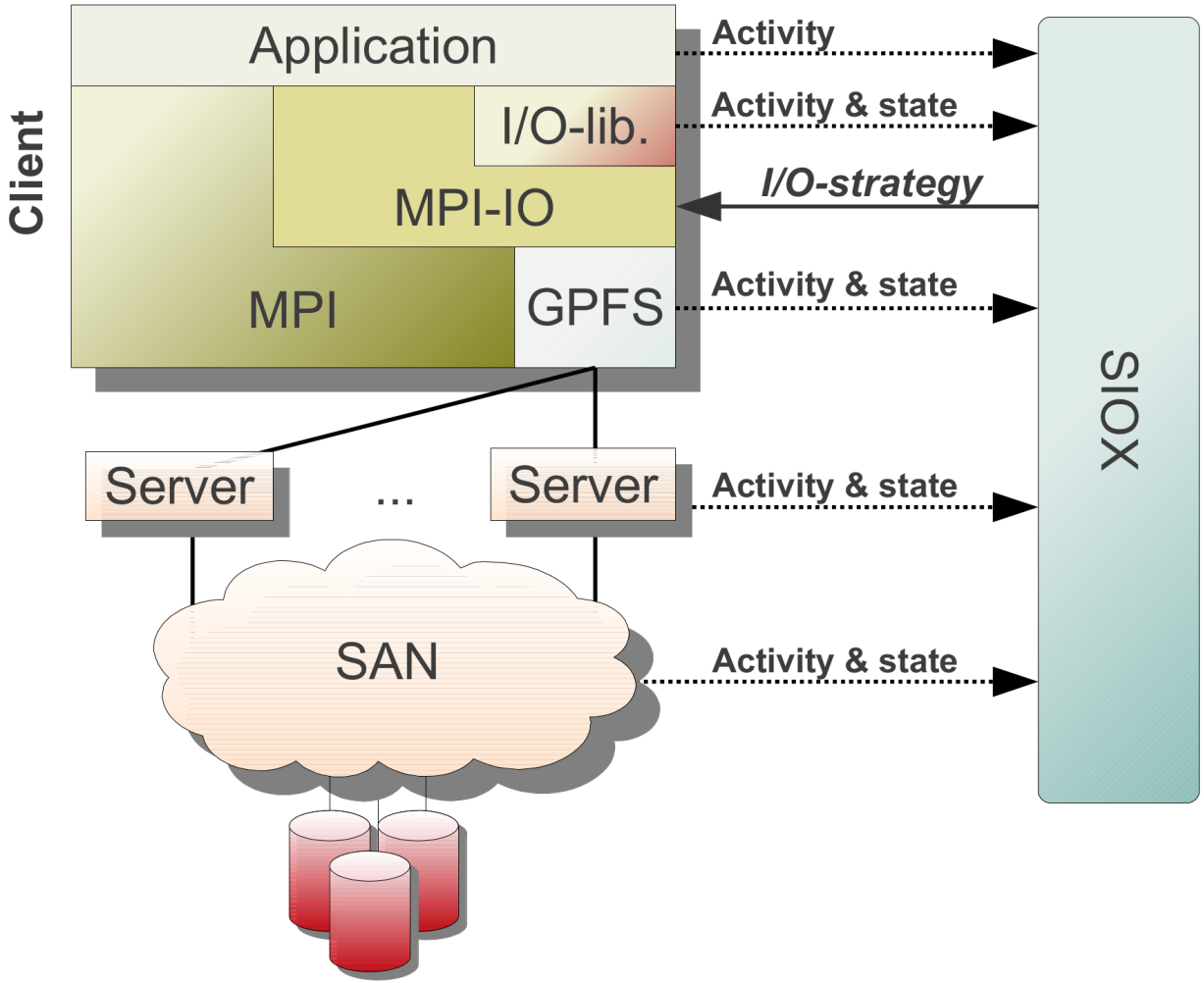
Ziele von SIOX

- Monitoring von E/A-Leistung im Produktivsystem
- Automatische Bewertung beobachteter Leistung
- Aufdeckung unbekannter Zusammenhänge
- Erlernen steuerbarer Optimierungen

Kooperationspartner

- Universität Hamburg, HLRS Stuttgart, ZIH Dresden, EIOW

Abstrakte Funktionsweise



Integration von SIOX in den E/A-Stack

Simulation von parallelen Anwendungen

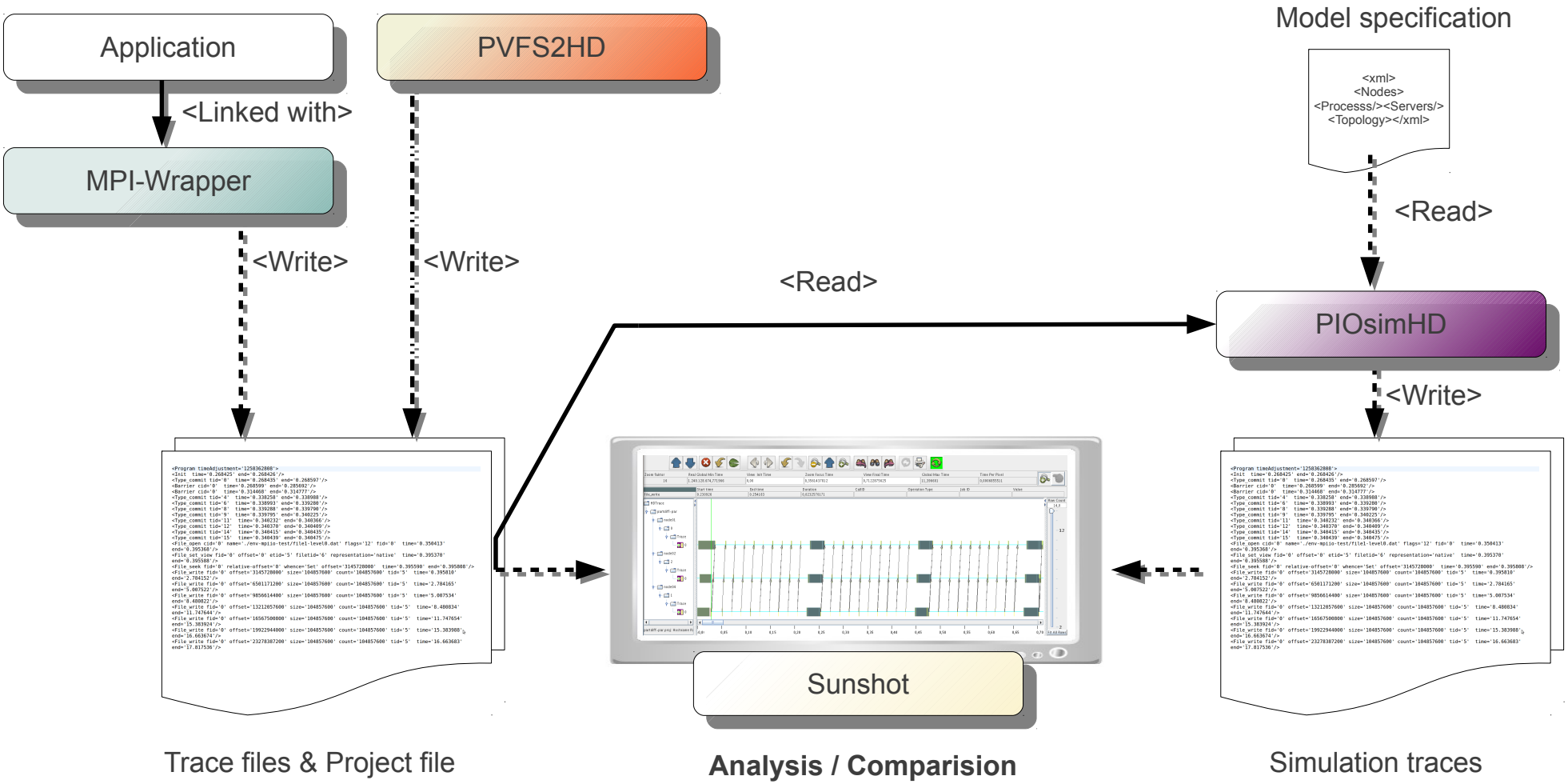
Ziele der virtuellen Forschungsumgebung PIOsimHD

- Lokalisierung von Leistungsengpässen und Ursachen
- Extrapolation von Leistungsfähigkeit für künftige Systeme
- Evaluation und Optimierungen des E/A-Pfades und MPI

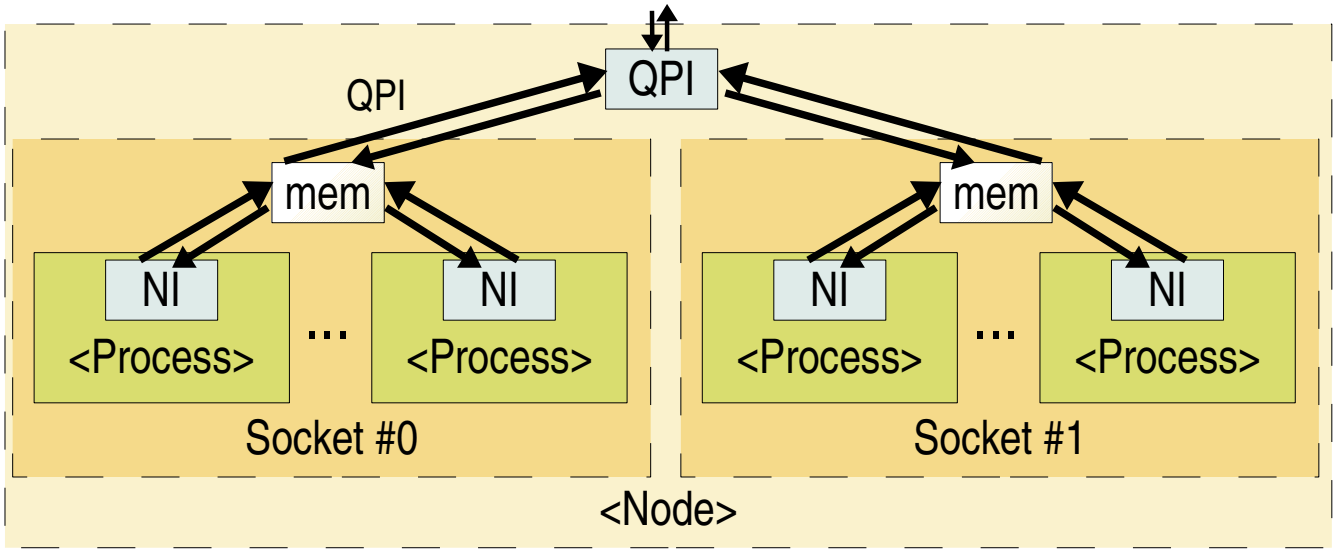
Funktionsübersicht

- Modularer ereignisorientierter Simulator
- Berücksichtigt elementare Hardware- und Software-Charakteristika
- Trace/Replay von realen Anwendungen
- Visualisierung mit Sunshot

Trace/Replay und Leistungsbewertung

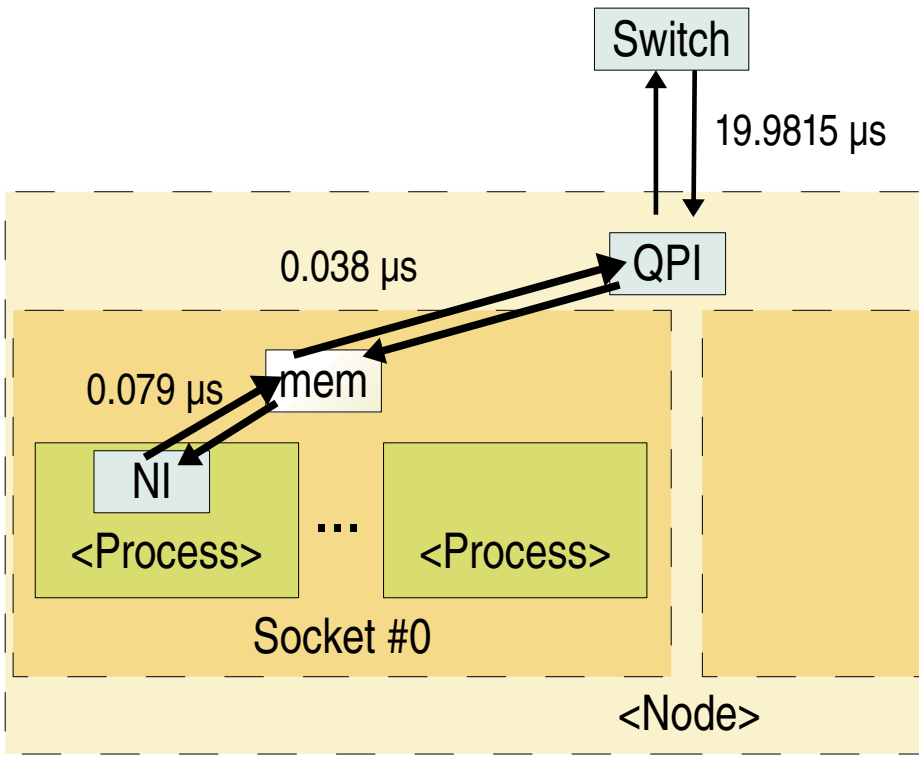


Beispielhafte Netzwerktopologie

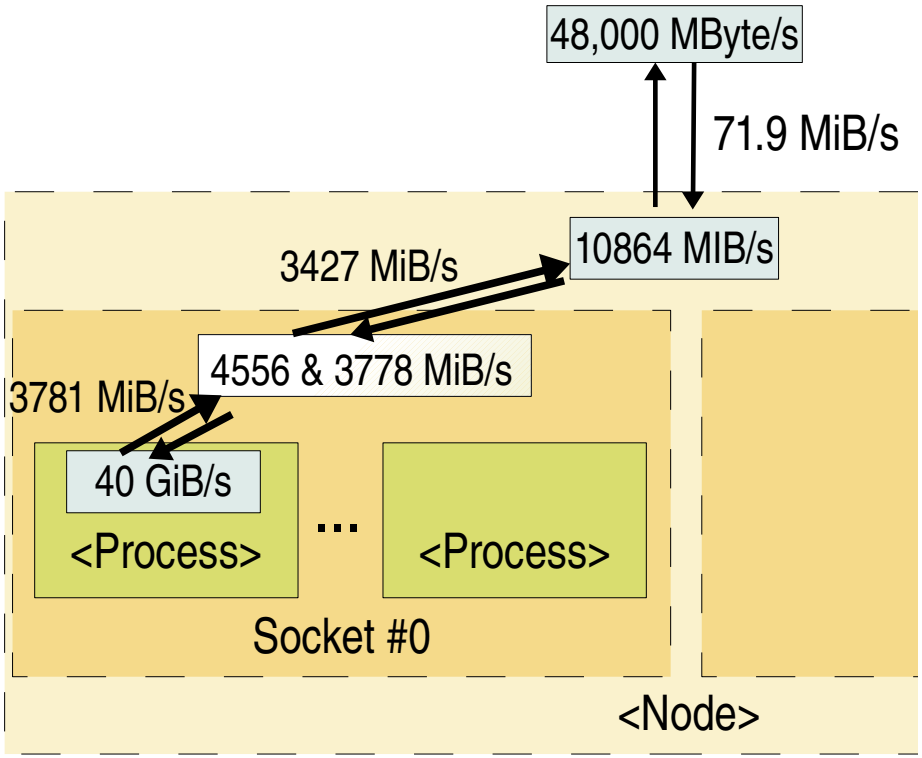


Modell für ein Dual-Sockel Westmere System

Beispielhafte Netzwerkcharakteristika



(a) Latenz



(b) Durchsatz

Validierung und Leistungsbewertung

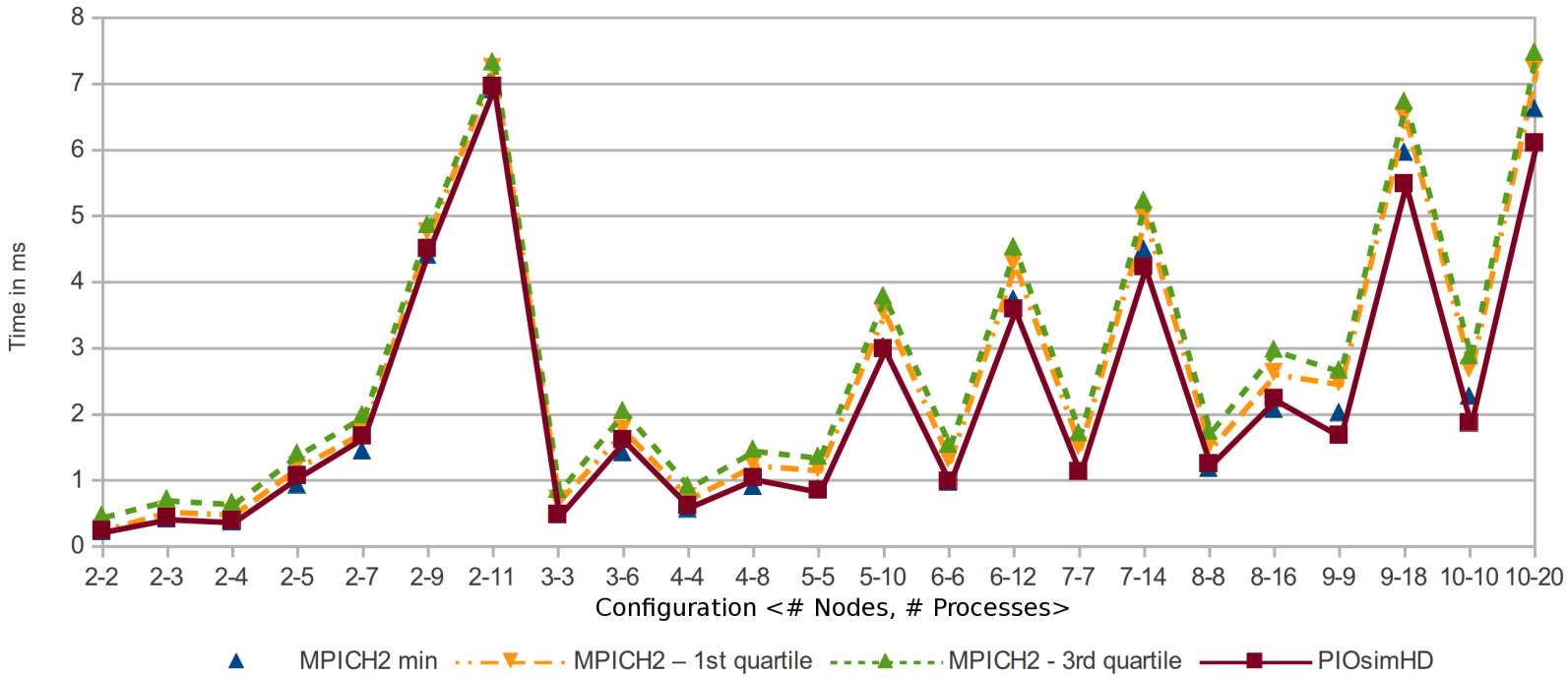
Szenarien

- Charakterisierung der Hardware und Modellbildung
- Punkt-zu-Punkt-Kommunikationsmuster
- Kollektive Operationen
  - Trace/Replay von Punkt-zu-Punkt-Operationen
- Parallele E/A
- Paralleles Programm: Jacobi-PDE-Löser

Fazit

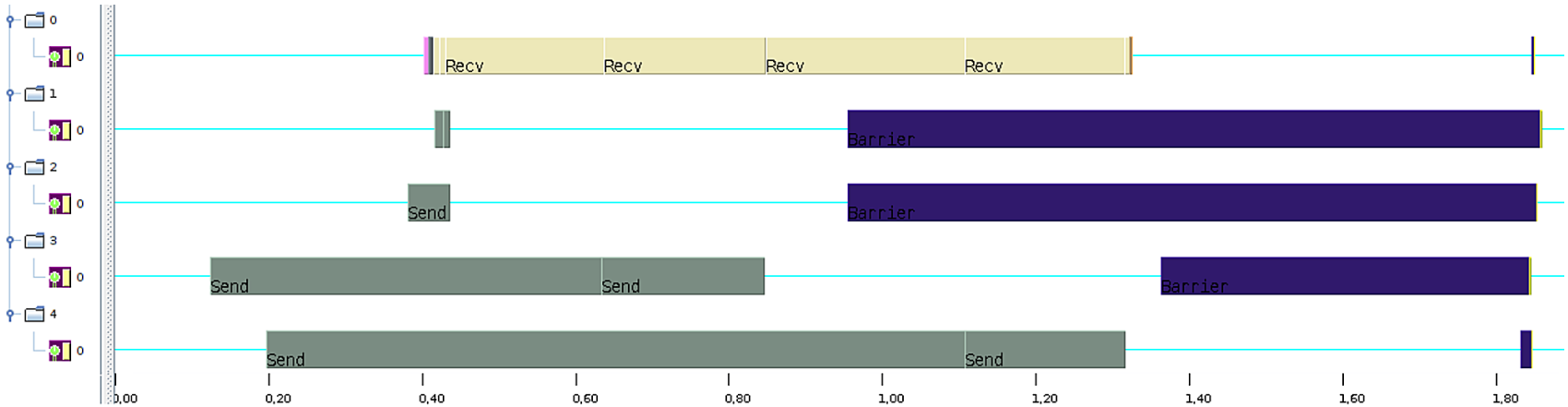
- Simulationsmodell approximiert Verhalten meist sehr gut
- Einige Ineffizienzen in System, MPI und PVFS wurden gefunden

Approximation eines kollektiven Algorithmus

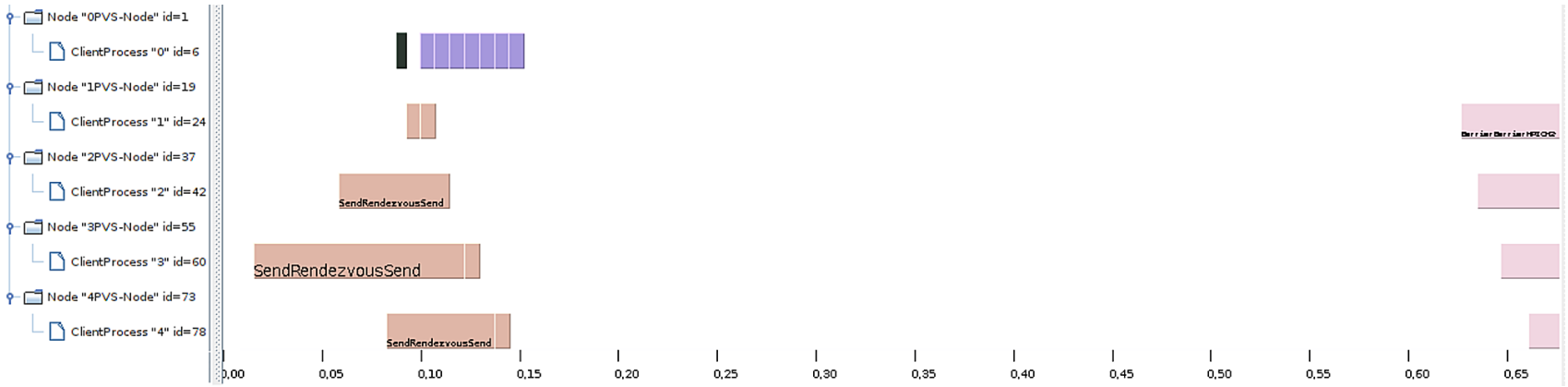


`MPI_Allgather()` für 10 KiB Daten

Aufdeckung von Ineffizienzen mit Sunshot



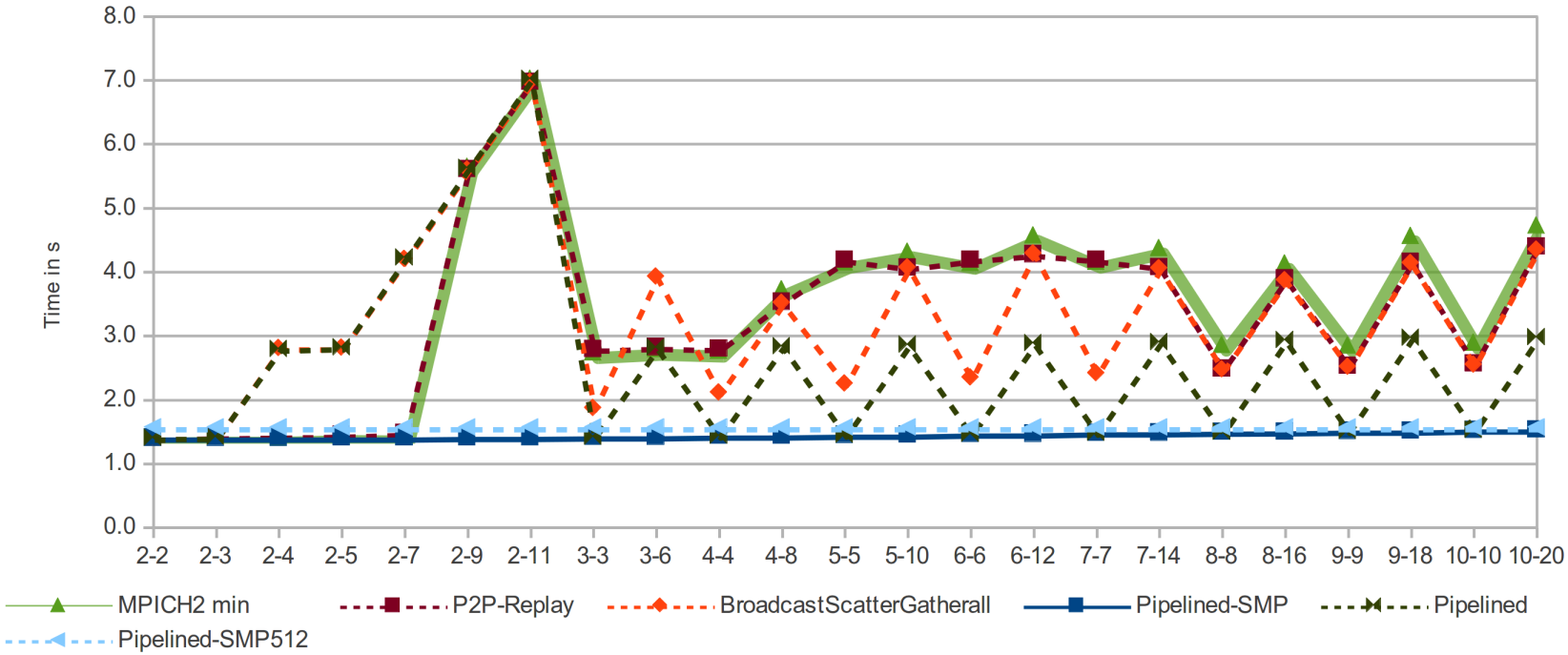
(a) Beobachtung



(b) Simulation

Endphase eines Jacobi-PDE-Lösers

Alternative kollektive Algorithmen



(a) Inter-Node Kommunikation (1)

`MPI_Bcast()`, 100 MiB Daten

Software Engineering in der Wissenschaft

Aktivitäten

- Interviews von Wissenschaftlern im Rahmen von Seminaren
- Erstellung von einer Online-Umfrage
  - Kentnisse von Werkzeugen und Nutzung (Testen, Leistungsanalyse, ...)
  - Zufriedenheit im Status-Quo

Erkenntnisse

- Wissenschaftler haben oft wenig SWE-Kentnisse
- Vergleichbare Konzepte und Werkzeuge werden „neu“ erfunden

$\Rightarrow$ Informatik hat Potential um Softwareentwicklung zu verbessern!

Energieverbrauch von Datenspeicherung

TEFL – **T**otal **E**nergy for the **F**ile **L**ife cycle

Ansatz

- Erfassung der Zugriffstatistiken und mittlerer Dateigröße
- Modell für aktiven und passiver Energieverbrauch der Speichersysteme
- Implementierungsvorschlag mit „Extended Attributes“

Beispielszenario

- 1 TByte Daten schreiben, 5 Jahre aufbewahren, 4 mal lesen
- Energiekosten
  - Online-Speicher: 211 €
  - Online-Speicher + Band: 1.3 €

Kompression von Klimadaten (im ICOMEX Projekt)

Ansatz

- Untersuchung verschiedener verlustloser Kompressionsalgorithmen
- Entwicklung von Filtern zur Ausnutzung von glatten Daten
- Aktuell: Verlustbehaftete Kompression und tolerierbare Fehler

Betriebswirtschaftliche Frage

Lohnt sich der zusätzliche Aufwand für die Kompression?

Fazit

- Dateigrößenreduktion konnte um weitere 14% verbessert werden
- Verlustlose Kompression spart 50% Medien
  - DKRZ: 5 PB Daten/Jahr $\Rightarrow$ 5600 € Server spart 40.000 € an Medienkosten
- Motivation für flexible Kompressionsalgorithmen in HPSS?

Erfassung von Nutzungsstatistiken im RZ

Ziel: Besseres Verständnis der Systemnutzung

Anwendung

- Fakten für den Dialog mit Benutzern
- Priorisierung von Analyse/Optimierung des Systems
- Anforderungen an künftige Supercomputer

Erfassung von Nutzungstatistiken im RZ (2)

Rechen-Jobs

- Systematische Erfassung von Programmstatistiken
- Auswertung: Knotenbedarf, Laufzeiten, Flop/s, Arbeitsspeicher

Dateisystem

- 4 PByte Work/Home: 160 Millionen Dateien
- Nutzung von Dateiformaten
  - TAR: einzelne Benutzer bis zu 20% Kapazität (wg. HPSS)
  - NetCDF: 17% der Dateien und 34% der Kapazität
  - HDF5: 0.2% der Dateien
- Größenverteilung von Dateien
  - Hochrechnung auf 60 Petabyte (DKRZ HLRE3)
  - 800 Mio Dateien $\leq$ 8 KByte $\Rightarrow$ 6 TByte SSDs
- Redundanz: 5% Dateiduplikate, 20% Inhaltsduplikation (CDC)

Zusammenfassung

- Hochleistungsrechnen ist ein spannendes Forschungsfeld
- Aktuelle Systeme arbeiten oft nicht optimal
- Supercomputer sind komplexe Systeme
- Die ganzheitliche Betrachtung von Hardware, Software und Nutzung ist für eine nachhaltige Optimierung notwendig
- Analyse bildet die Grundlage für eine Bewertung
- Simulation ist ein Zugang zur Analyse und Bewertung
- Nutzungsverhalten bestimmt Optimierungspotential

Forschungspläne (1)

Rechenzentrumsnahe Forschung

- Systemweite Analyse von Nutzungs- und Systemverhalten
- Expertensystem zur Leistungsbewertung von Anwendungen
- Automatische Lokalisierung von Engpässen in MPI und System
- Studien zur wissenschaftlichen Softwareentwicklung
- Betriebswirtschaftliche Betrachtungen

Forschungspläne (2)

Analyse und Optimierung

- Benchmark für Replay von E/A- und Kommunikationsmustern
- Portal für Benchmarks, Lasten und Ergebnisse
- Optimale kollektive Algorithmen für Clustercomputer
- System-aware MPI – Nutzung von Systemcharakteristika

Parallele Ein-/Ausgabe

- Analyse und Optimierung von E/A-Middleware und Dateisystemen
- Benutzerorientierte E/A-Schnittstelle und Semantik
- Simulation von paraller E/A auf verschiedenen Abstraktionsschichten

Backup Folien

Persönliche Einschätzung

Steigender Bedarf an Rechenleistung und Effizienz führt zu

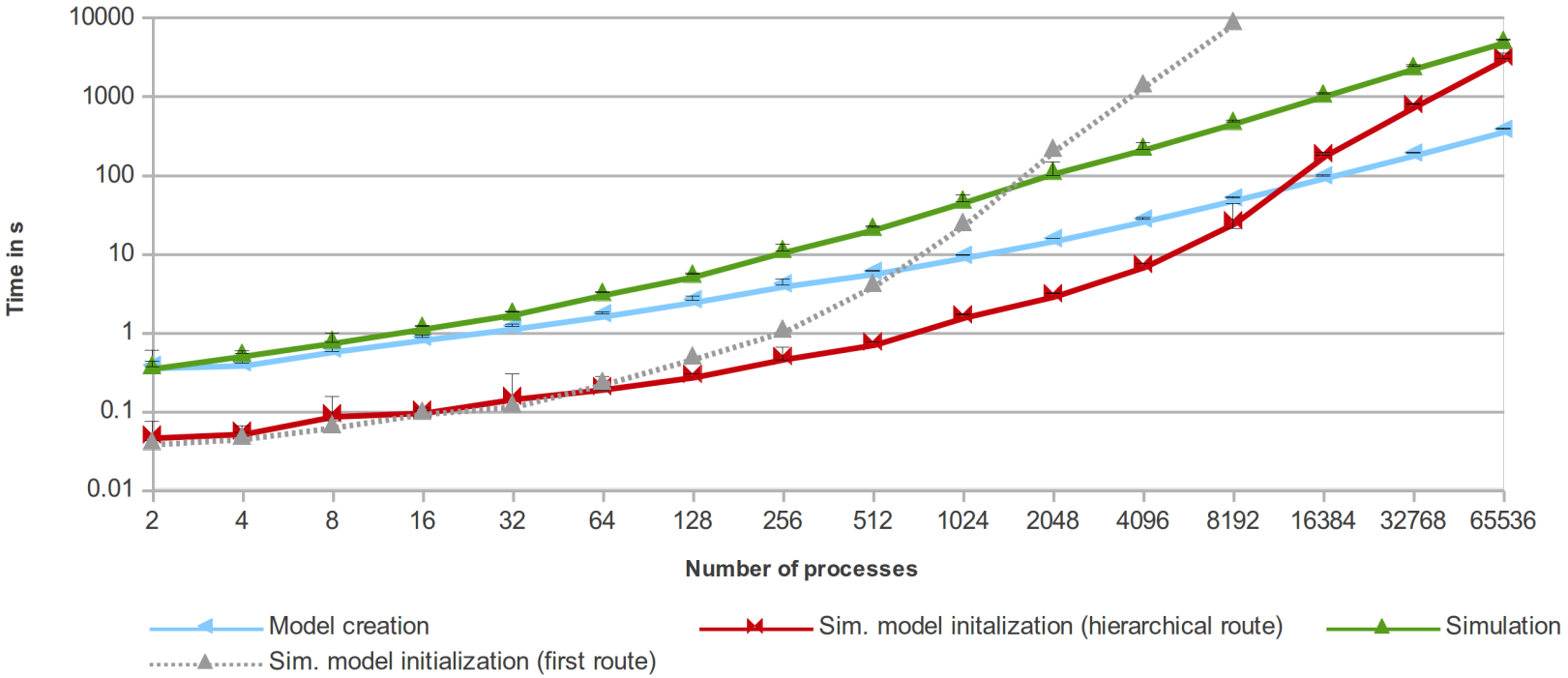
- Heterogenität von Systemen
- Mehr Programmiermodellen
- Spezialisierungen
- Interdisziplinärer Zusammenarbeit

Steigende Kosten führen zu

- Systematischer Optimierung von Programmen und Hardware
- Rechenzentrumweite Analyse von Benutzer- und Systemverhalten

$\Rightarrow$ Verständnis von Anwendungs- und Systemverhalten wird wichtiger!

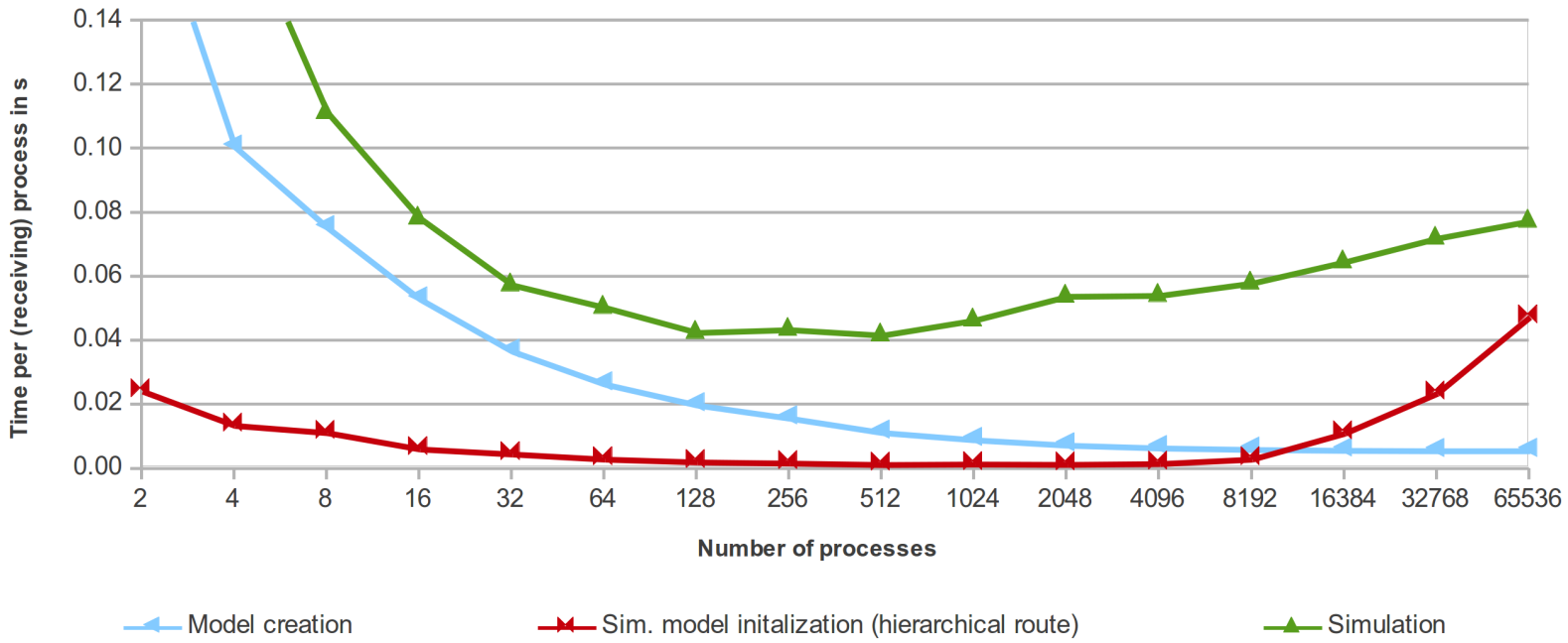
Leistungsfähigkeit des Simulators



(a) Laufzeit

Simulation von `MPI_Bcast()`

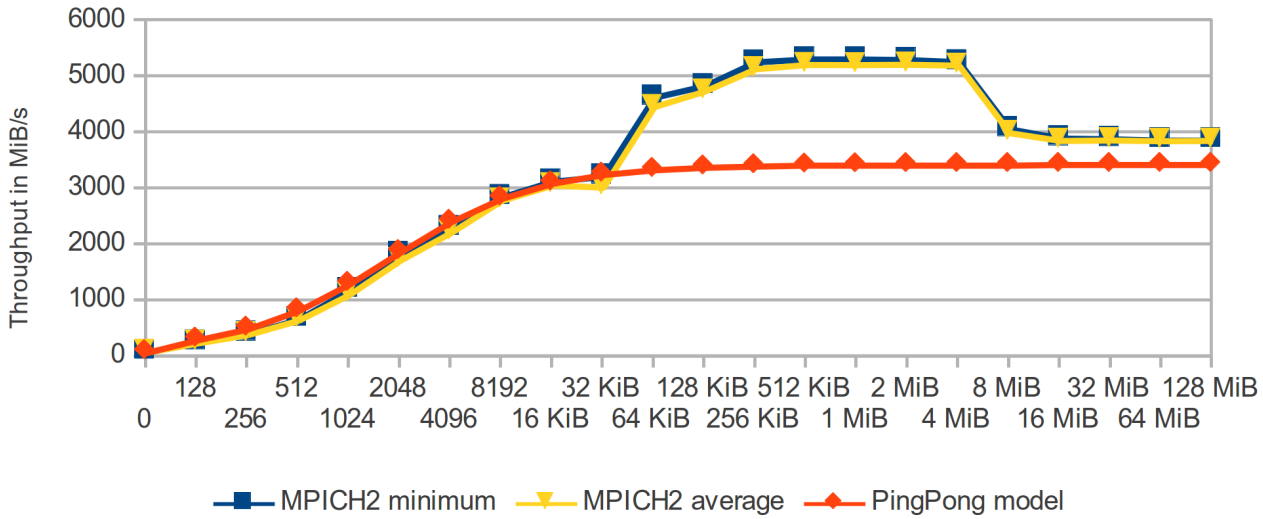
Leistungsfähigkeit des Simulators

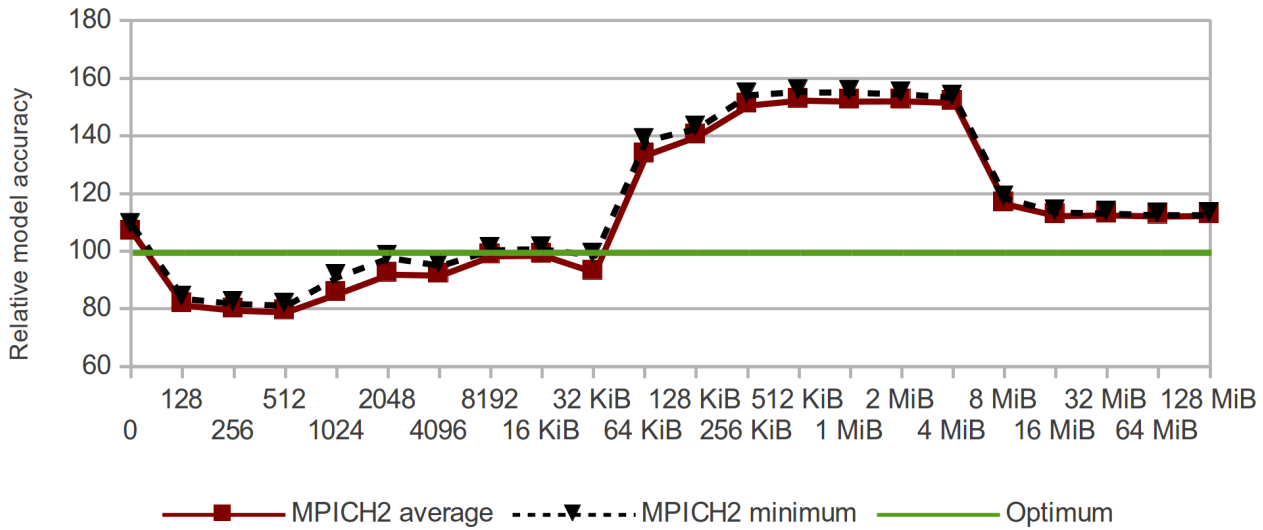


(b) Skalierbarkeit

Simulation von `MPI_Bcast()`

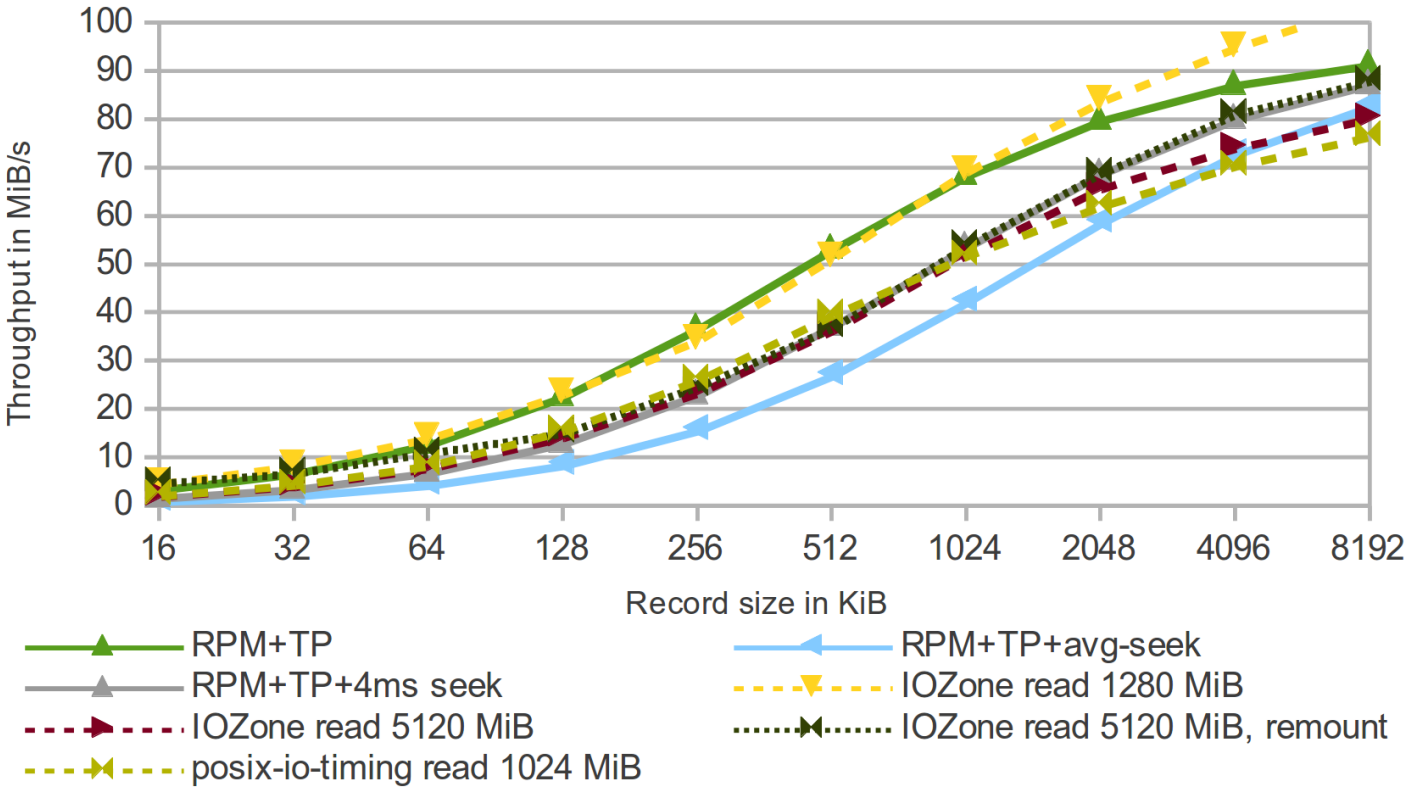
Verifikation (Qualification) des Netzwerkmodels





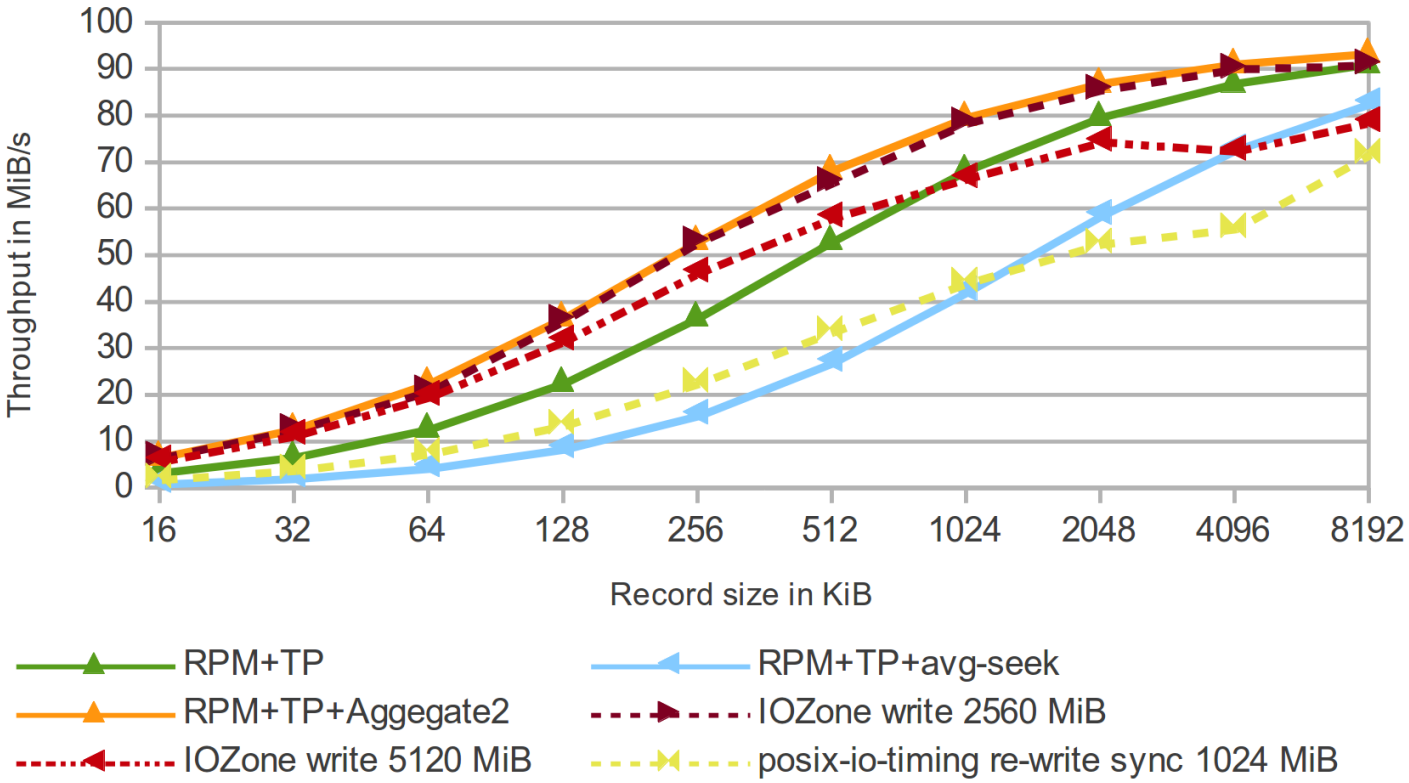
Inter-socket Kommunikationsleistung

Verifikation des E/A-Models für eine Festplatte



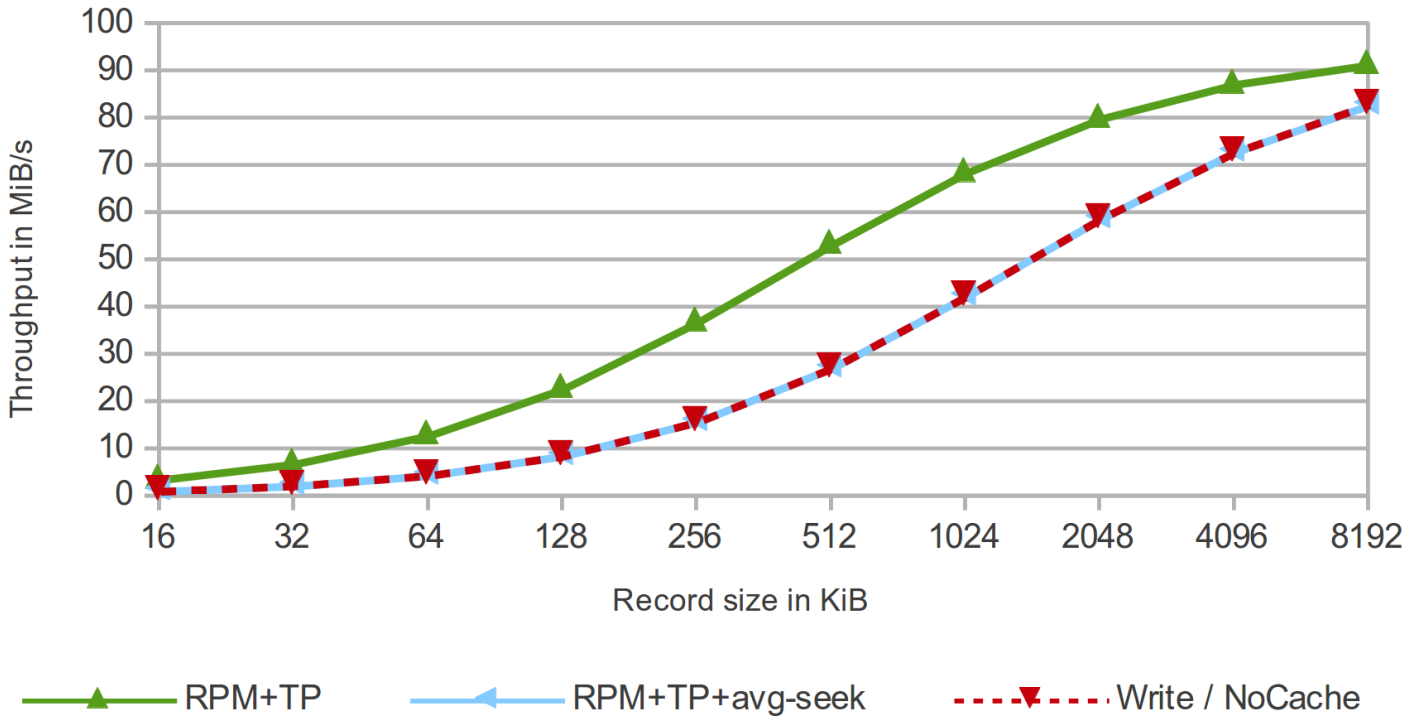
Zufällige Lese-Zugriffe

Verifikation des E/A-Models für eine Festplatte (2)



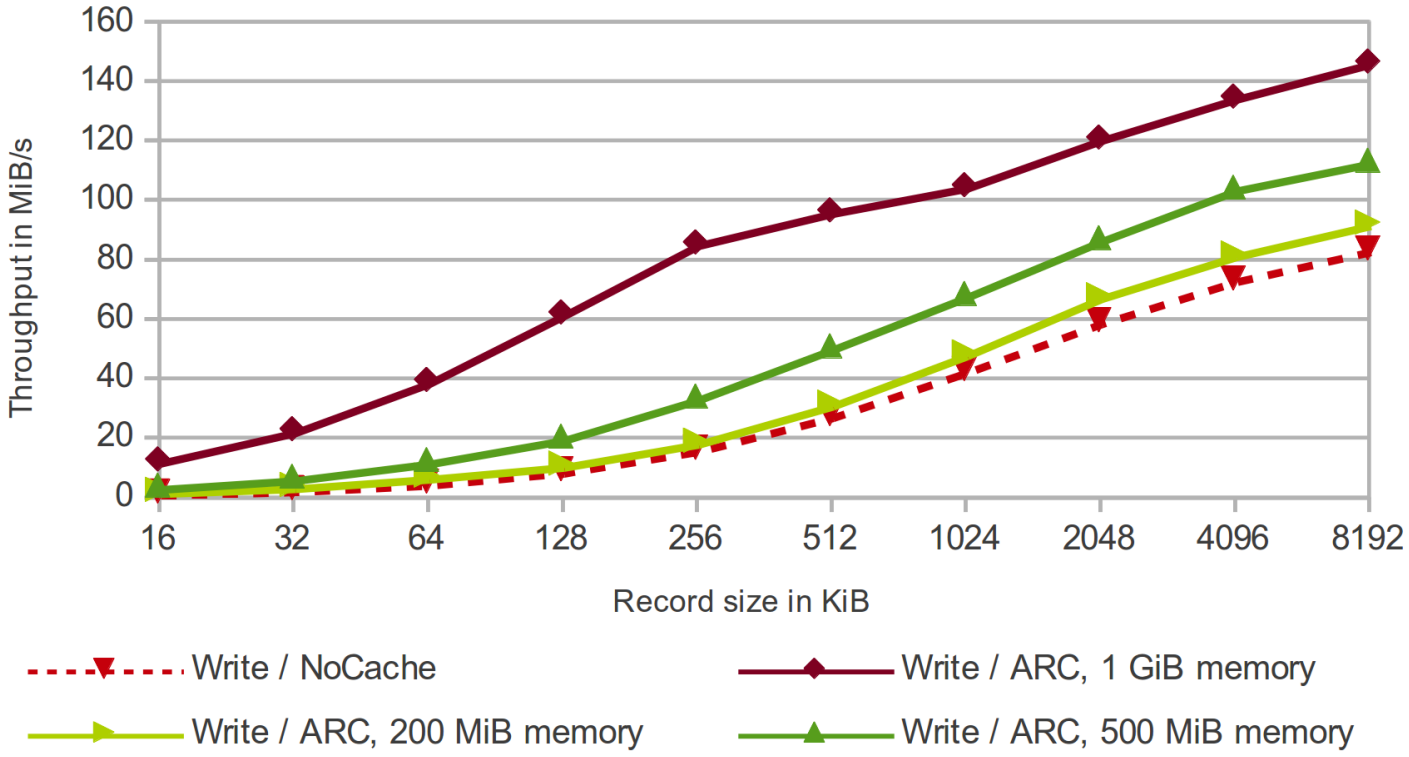
Zufällige Schreib-Zugriffe

Verifikation der simulierten Festplatte



Zufällige Schreib-Zugriffe ohne Cache

Verifikation der simulierten Festplatte

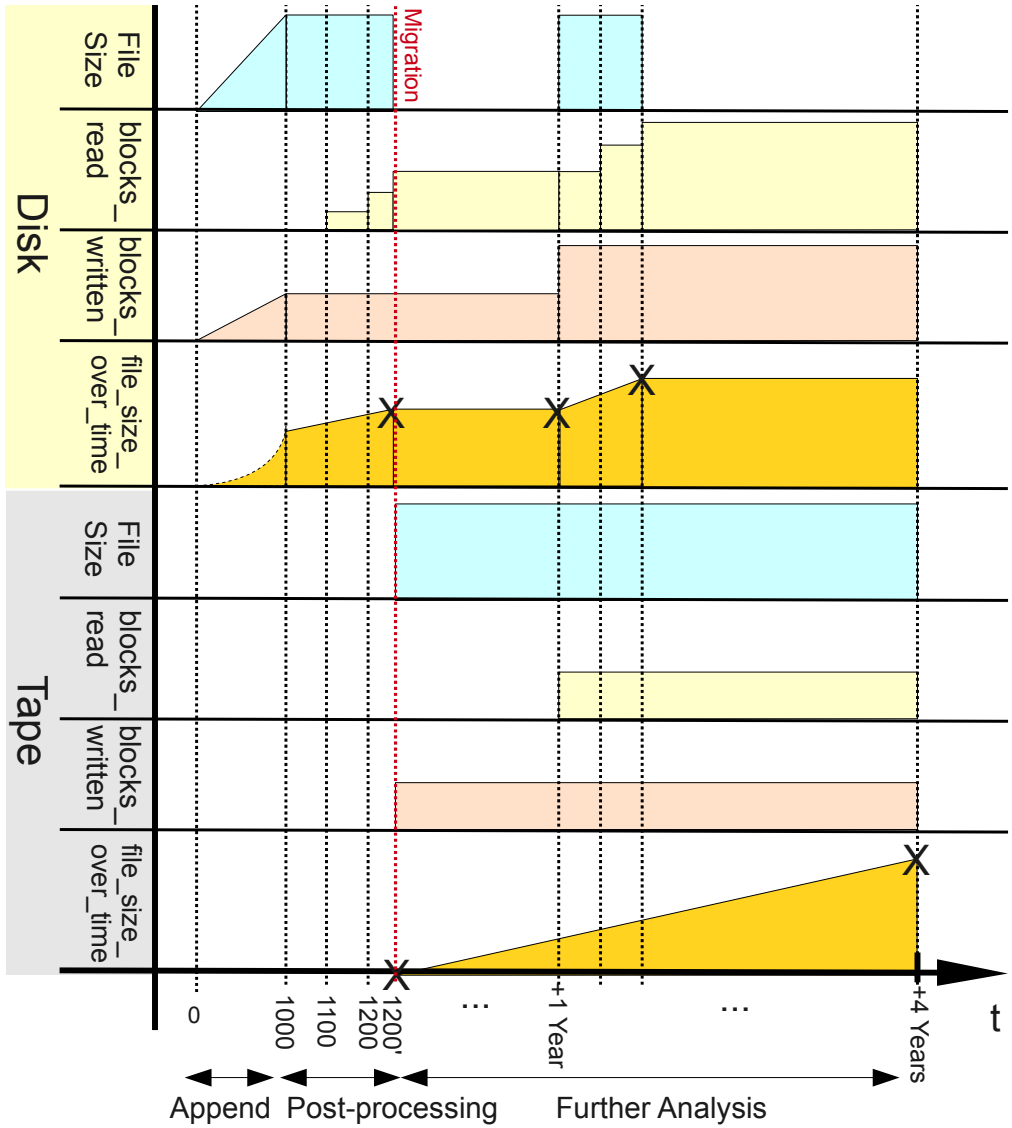


Zufällige Schreib-Zugriffe mit AggregationReorderCache

Notwendige „Extended Attributes“

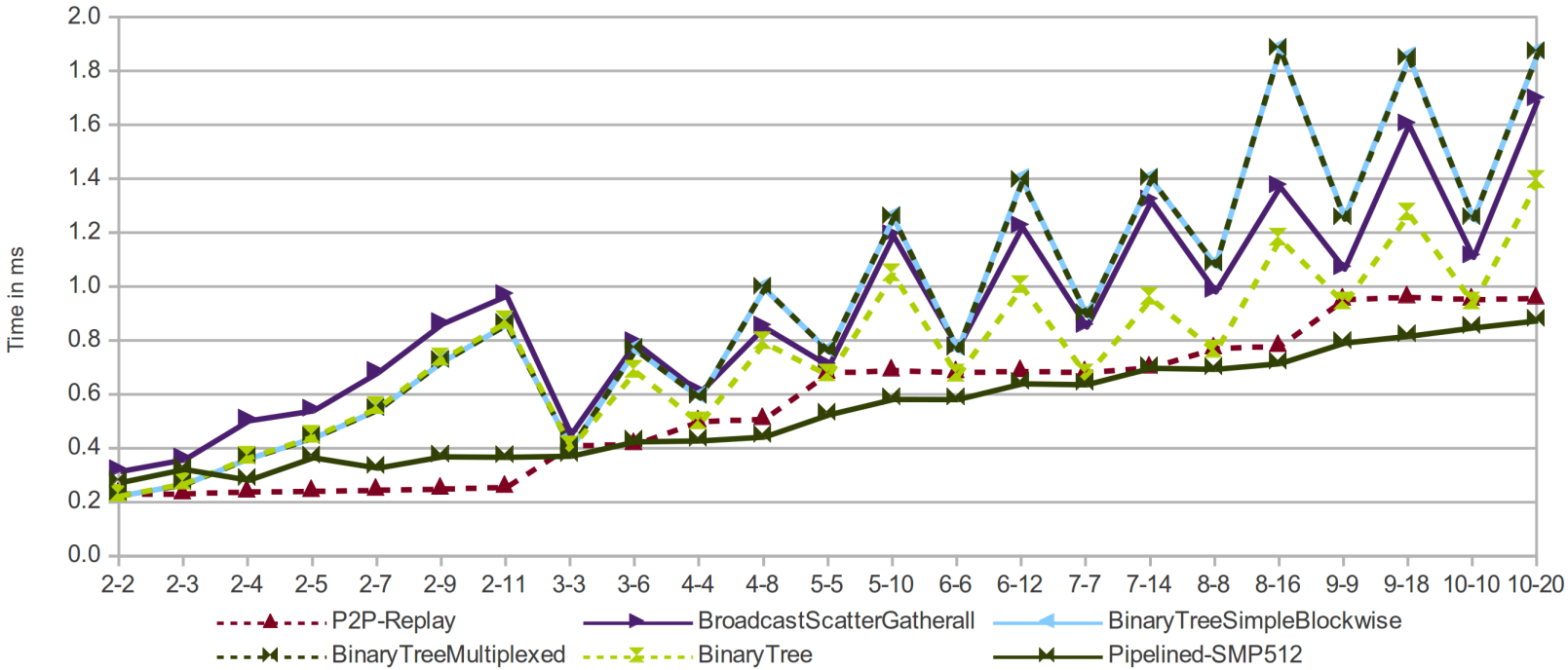
system.**iocount_read**	= 10	
system.**iocount_write**	= 50	
system.**blocks_read**	= 1000	
system.**blocks_written**	= 1000	
system.**file_size_over_time**	= 200000	*(Byte * Seconds)*
system.**last_file_size_update**	= 2010-05-12 17:00:00.01	*(timestamp)*

Extended Attributes und Beispielwerte



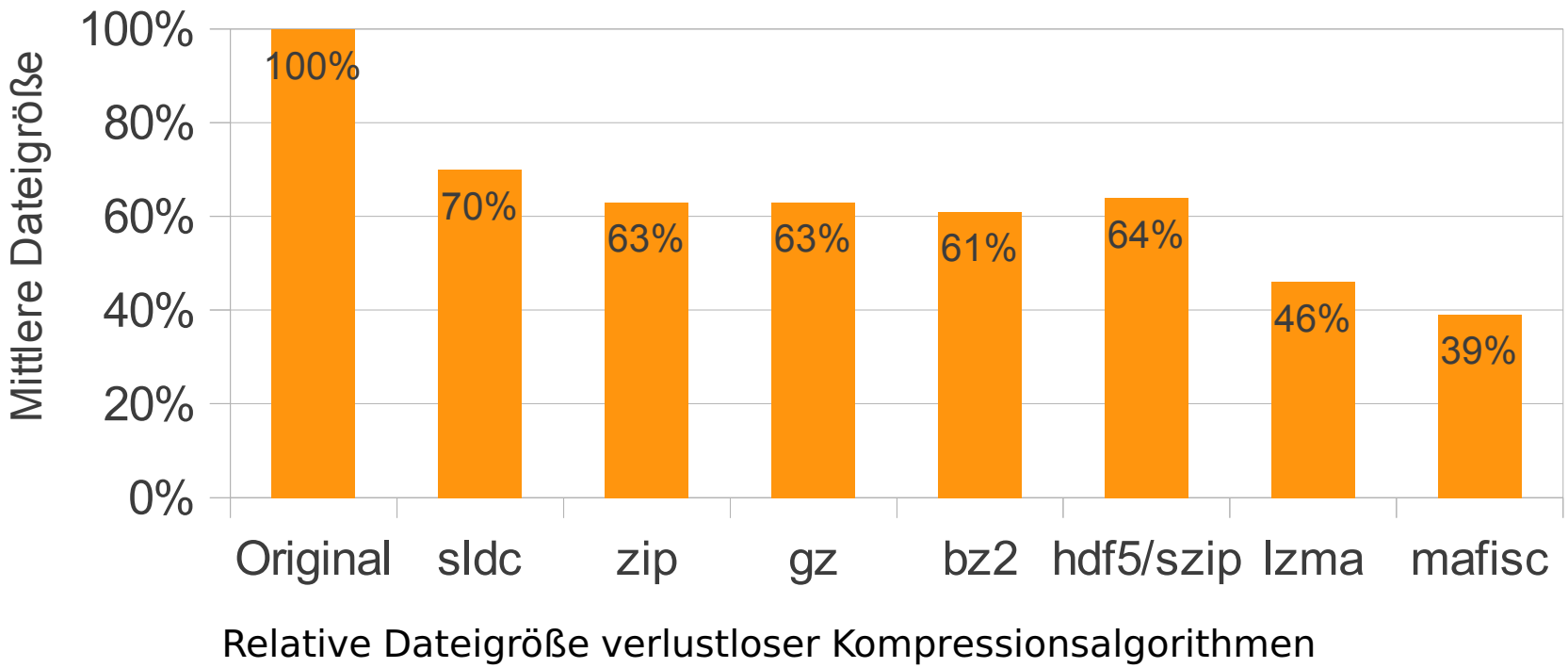
Veränderung der Dateiattribute für eine Migration zwischen Dateisystem und Band (qualitativ)

Alternative kollektive Algorithmen

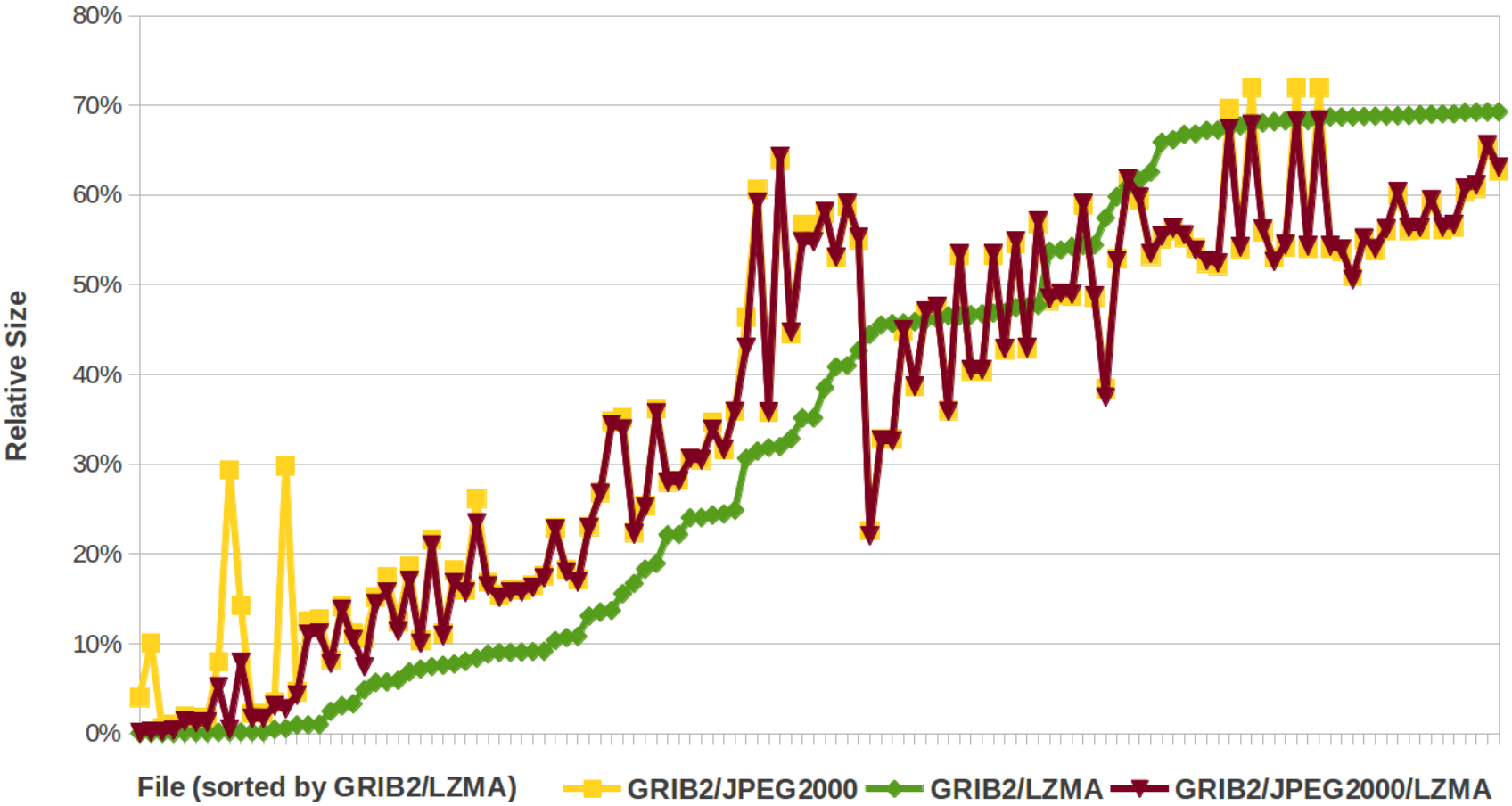


(a) Inter-Node Kommunikation

`MPI_Bcast()`, 10 KiB Daten

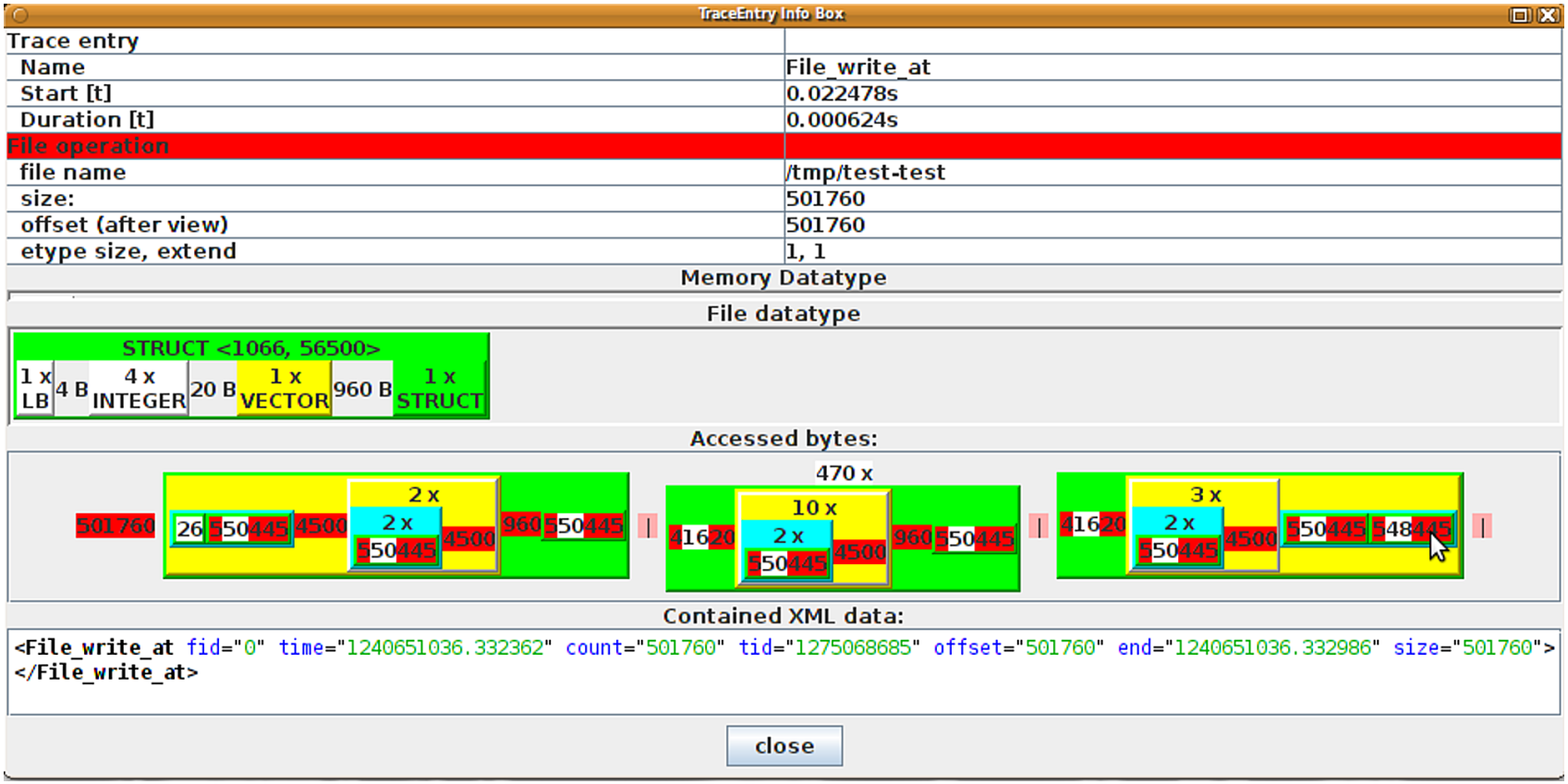

Mittlere Dateigröße
100%
80%
60%
40%
20%
0%
100%
70%
63%
63%
61%
64%
46%
39%
Original
sldc
zip
gz
bz2
hdf5/szip
lzma
mafisc
Relative Dateigröße verlustloser Kompressionsalgorithmen

Verlustbehaftete Kompression von Klimadaten mit GRIB



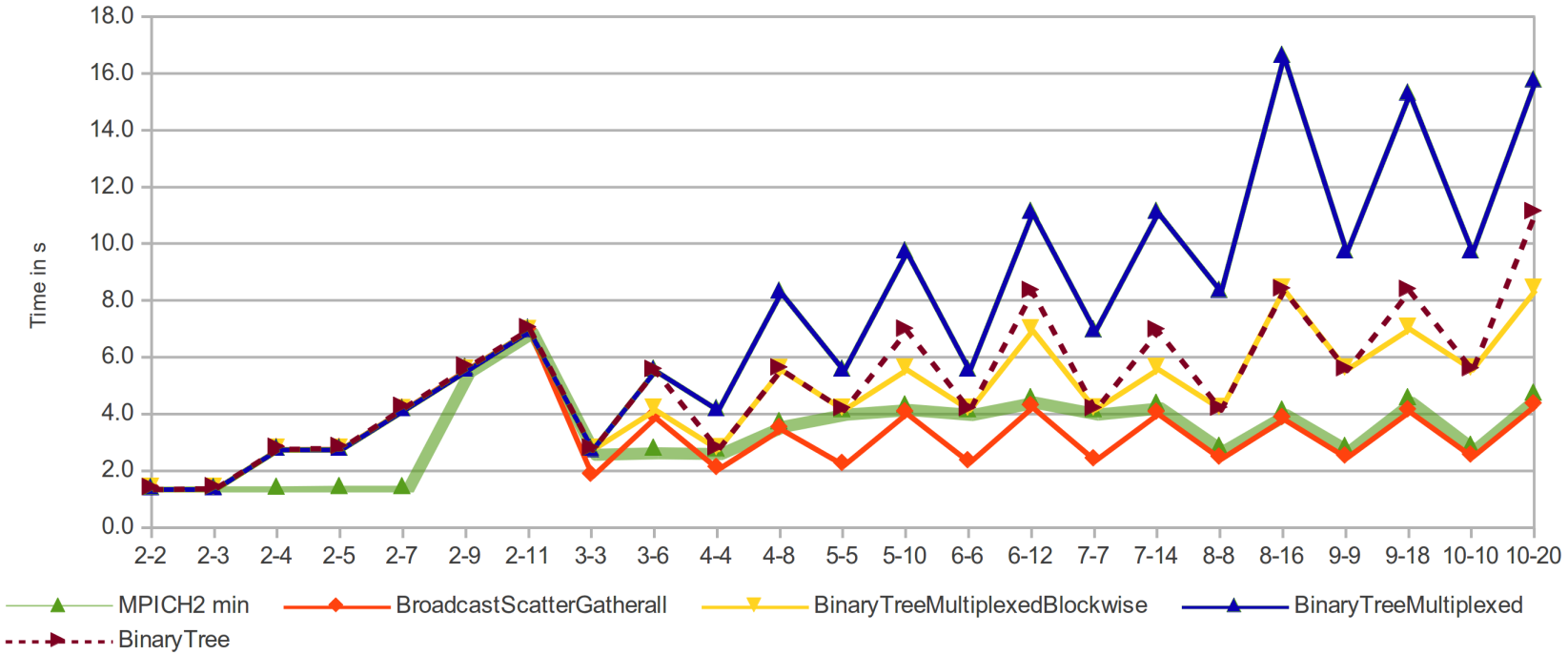
Relative Dateigröße zusätzlicher Kompression von 22 bit GRIB kodierten ECHAM Daten

MPI-IO Zugriffe



Visualisierung von Datentyp und zugegriffene Datei-Bytes

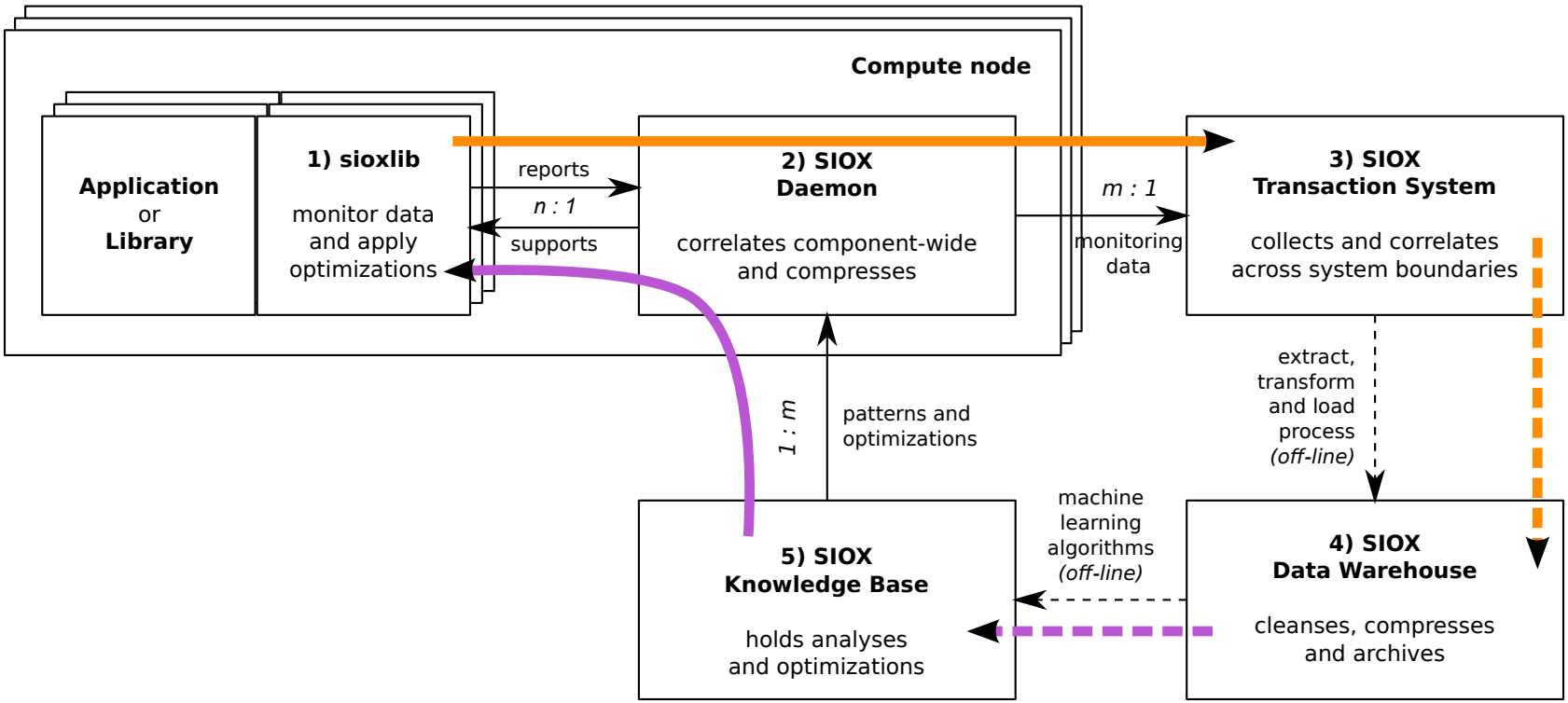
Alternative kollektive Algorithmen



(b) Inter-Node Kommunikation (2)

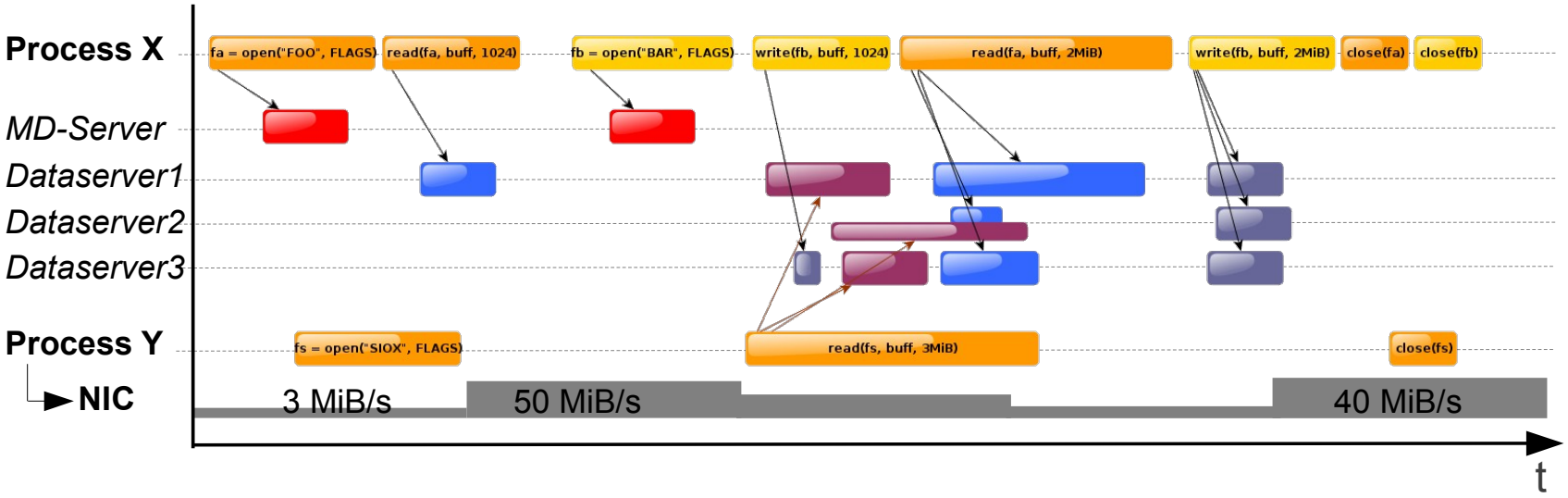
`MPI_Bcast()`, 100 MiB Daten

Architekturentwurf



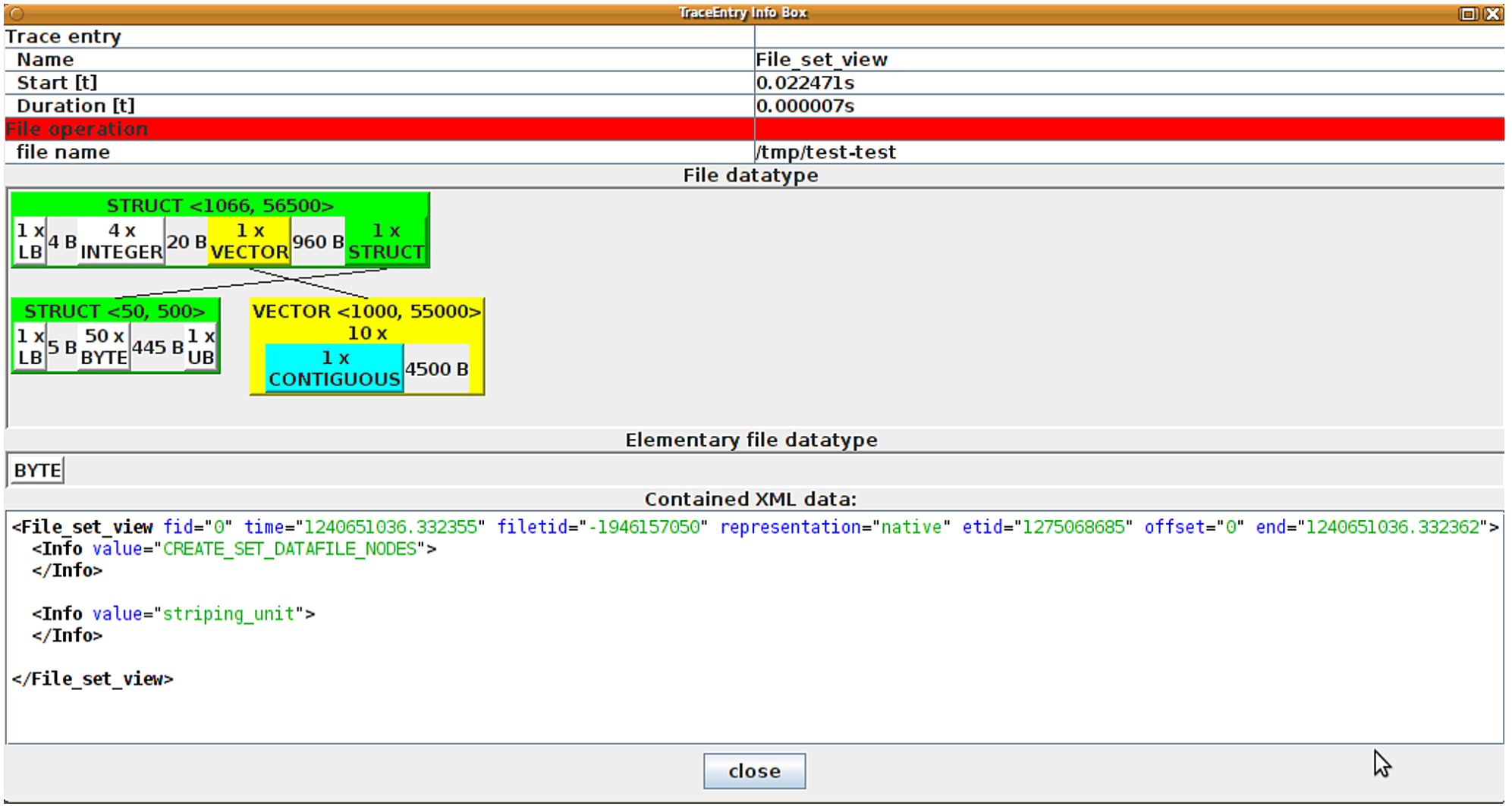
Komponenten von SIOX

Systemverhalten beim Dateizugriff



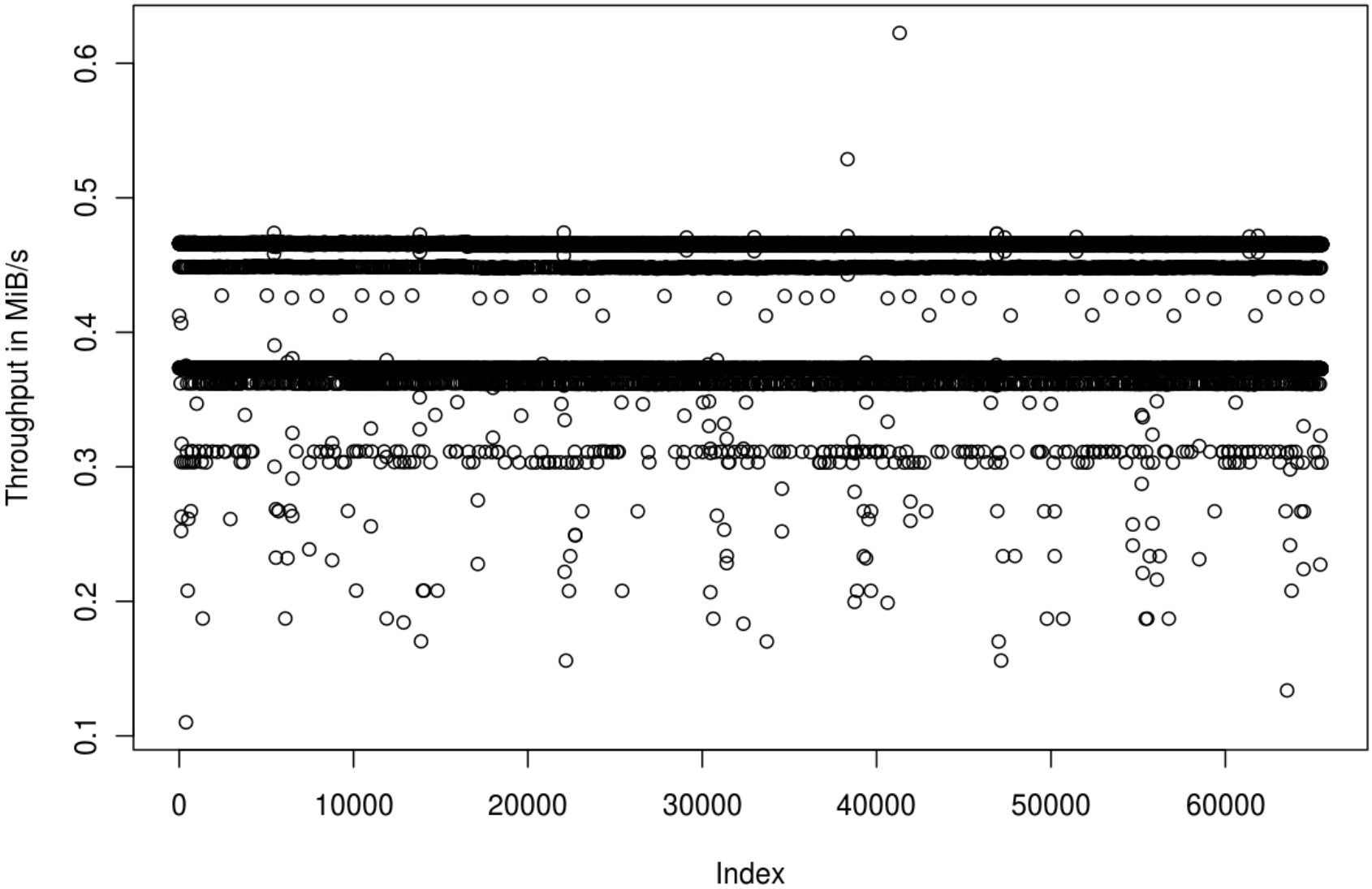
Clientaktivitäten und ausgelöstes Serververhalten

Benutzerdefinierte Datentypen in MPI



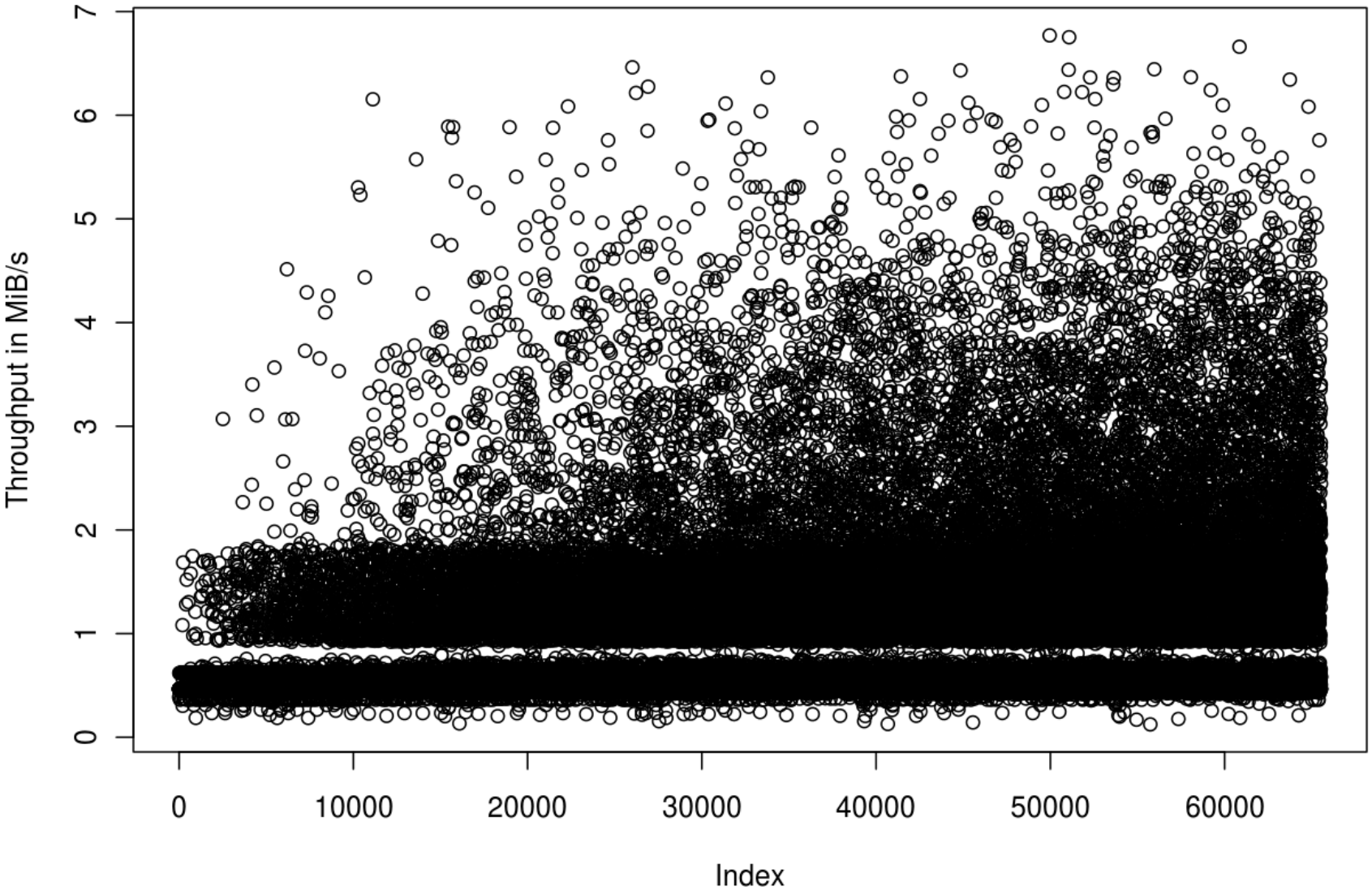
Auffaltbare Repräsentation von MPI-Datentypen

E/A-Leistungsfähigkeit



Sequentielles Schreiben von 16 KiB Blöcken (*O_DIRECT*, *O_SYNC*)

E/A-Leistungsfähigkeit



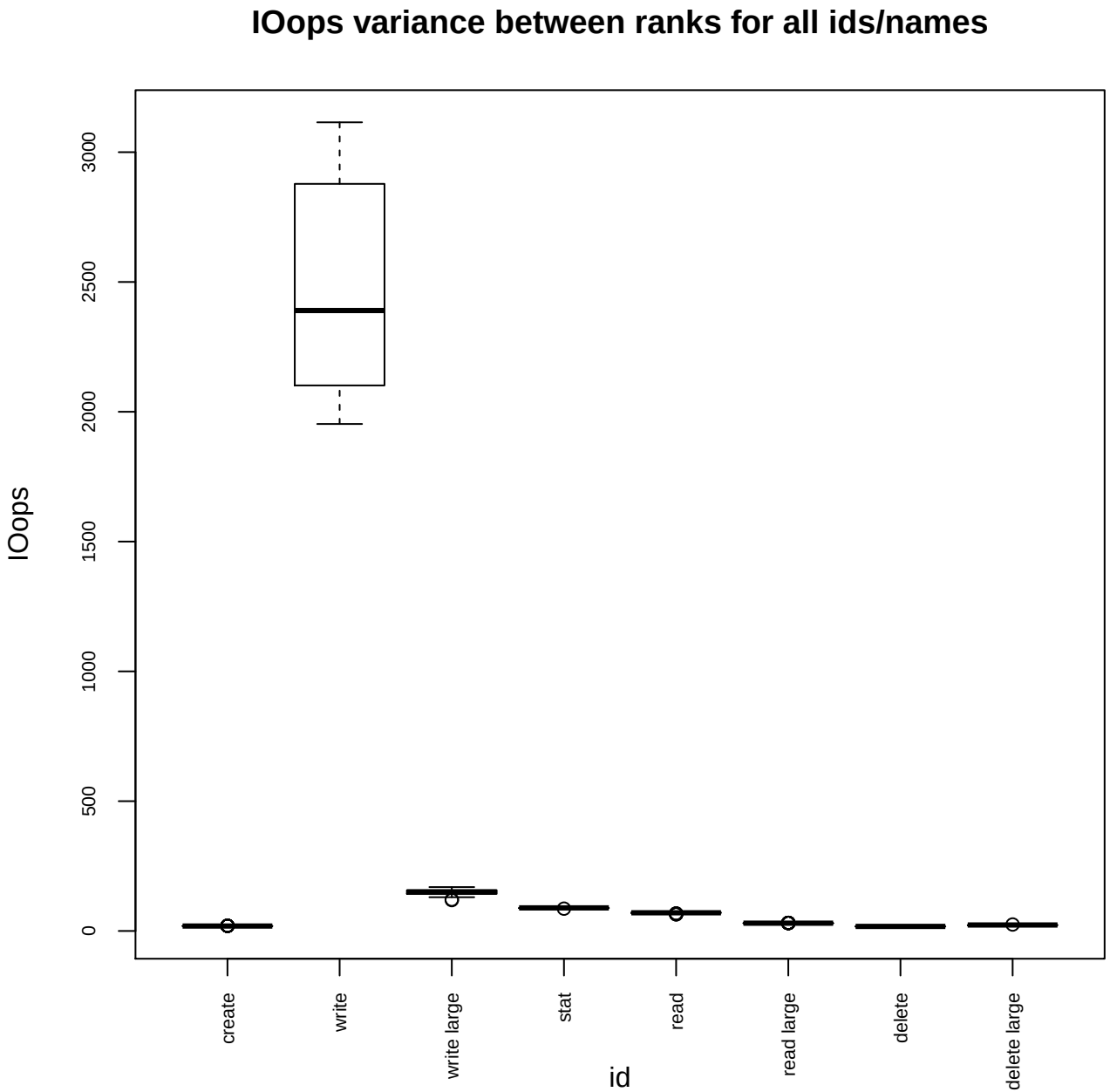
Zufälliges Schreiben von 16 KiB Blöcken (*O_DIRECT*, *O_SYNC*)

Automatische Optimierung in SIOX

- Plugin-Architektur für beliebige Optimierungen
- Optimierungsentscheidung basiert auf
  - Aktivitätsmuster
  - Systemauslastung

- Beispiel für die Optimierung des E/A-Caches

Aktivitätsmuster	E/A-Cache
open()	4 MiB
write(size $<$ 2 KiB){5}	1 MiB
write(size $<$ 4 MiB){2}	20 MiB
write(size $\geq$ 100 MiB)	direct-write


IOops variance between ranks for all ids/names
IOops
0
500
1000
1500
2000
2500
3000
create
write
write large
stat
read
read large
delete
delete large
id