

Leistungsanalyse

Hochleistungs-Ein-/Ausgabe

Michael Kuhn

Wissenschaftliches Rechnen
Fachbereich Informatik
Universität Hamburg

2017-06-16



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

- 1 Leistungsanalyse
 - Orientierung
 - Einleitung
 - Leistungsmessung
 - Leistungsbewertung
 - Zusammenfassung

- 2 Quellen

Einleitung

- Leistungsanalyse ist ein schwieriges Thema
 - Software und Hardware immer komplexer
 - Viele Schichten und Interaktionen
- Analyse besteht aus zwei Teilen
 - Messung und Bewertung
- Messung erfasst aktuelle Leistungsfähigkeit
 - Richtig messen ist auch schwierig
- Bewertung erlaubt Vergleich mit möglicher Leistung
 - Insbesondere bei Beschaffungen etc.

Benchmarks

- Unmengen an Benchmarks verfügbar
 - Für alle möglichen Aspekte
 - CPU, RAM, Netzwerk etc.
- Für E/A ebenso
 - IOzone, Bonnie, Bonnie++, PostMark, b_eff_io, FLASH I/O und viele mehr
 - Vorgestellt: fio, IOR und mdtest
- Üblicherweise Erfassung eines bestimmten Zugriffsmusters
 - Daher viele verschiedene Benchmarks für viele verschiedene Anwendungsfälle



fio...

- Operationstypen
 - Lesen/schreiben/gemischt, sequentiell/zufällig
- Blockgröße
 - Sowohl einzelne Werte als auch Bereiche
- Datengröße
 - Wie viele Daten insgesamt zugegriffen werden sollen
- E/A-Engine
 - Synchron/asynchron, Memory Mapping, null
- Warteschlangentiefe
 - Für asynchrone E/A-Engines

fio...

- Ausrichtung
 - Beispielsweise zur Ausrichtung der E/A an Seitengrenzen
- Komprimierbar- und Deduplizierbarkeit
 - Aktuelle SSDs komprimieren die Daten transparent
- Durchsatzlimit
 - Nützlich um Hintergrundlast zu erzeugen
- Verifikation
 - Überprüfung ob die gelieferten Daten den vorher geschriebenen entsprechen

mdtest

- Datendurchsatz durch Benchmarks gut abgedeckt
 - Metadaten aber auch ein wichtiger Faktor
- mdtest nutzt MPI für parallelen Metadatenzugriff
 - Operationen werden mit POSIX-Schnittstelle durchgeführt
 - MPI-IO bietet keine ausreichende Funktionalität
- Aufgeteilt in drei Phasen
 - Erstellen, Status abrufen und Löschen
- Arbeitet mit hierarchischer Struktur

mdtest...

- Unterstützt ähnliche Funktionalitäten wie IOR
 - Prozessumordnung für Cache-Umgehung
 - Mehrere Wiederholungen
- Optional können Daten in Dateien geschrieben werden
 - Bei Bedarf auch synchronisiert werden
- Unterstützung für mehrere Wurzelverzeichnisse
 - Beispielsweise um mehrere Metadatenserver zu nutzen

Vergleich

- Vielzahl von Benchmarks für unterschiedlichste Anwendungsfälle
 - Sowohl Daten als auch Metadaten
- Ergebnisse allerdings nicht vergleichbar
 - Unterschiedliche Muster
 - Unterschiedliche Berechnung
 - Unterschiedliches Verhalten
- Unklarheiten bezüglich Ergebnissen
 - MB vs. MiB

Werkzeuge

- Benchmarks erlauben Erfassung der momentanen Leistung
 - Kein Aufschluss über Ursachen etc.
- Analyse und Optimierung erfordern Werkzeuge
 - Einblick in innere Vorgänge notwendig
 - Üblicherweise mit Hilfe von Tracing (Score-P)
- Manchmal abstrakte Betrachtung ausreichend
 - Grobe Übersicht über E/A-Verhalten (Darshan)

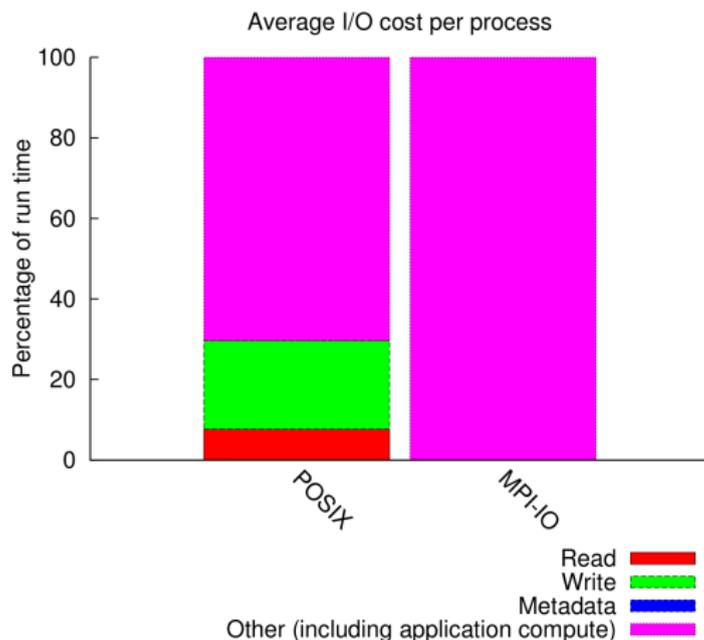
Darshan...

- Besteht aus zwei Teilen
 - Runtime und Werkzeuge
- Runtime erfasst Anwendungs-E/A
 - Spezifisch für eine MPI-Implementierung
 - Außerdem Optionen für Batch-Scheduler und gemeinsames Protokollverzeichnis
 - Compiler-Wrapper und Preload-Bibliothek `libdarshan.so`
- Werkzeuge analysieren aufgezeichnete Protokolle
 - `darshan-job-summary.pl`, `darshan-parser` etc.

Darshan...

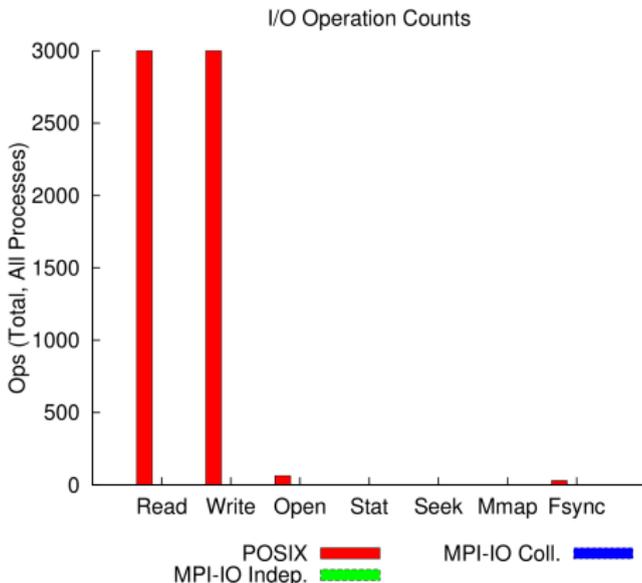
- MPI-parallelisierter POSIX-Benchmark mit zehn Prozessen
 - Erst Schreibphase, dann Lesephase
 - Durch Barriers getrennt
 - Blockgröße von 1 MiB
 - 100 Blöcke
 - Leeren des Caches zwischen Schreib- und Lesephase
 - `echo 3 > /proc/sys/vm/drop_caches`
 - `fsync` beim Schließen
 - Drei Wiederholungen

Darshan...



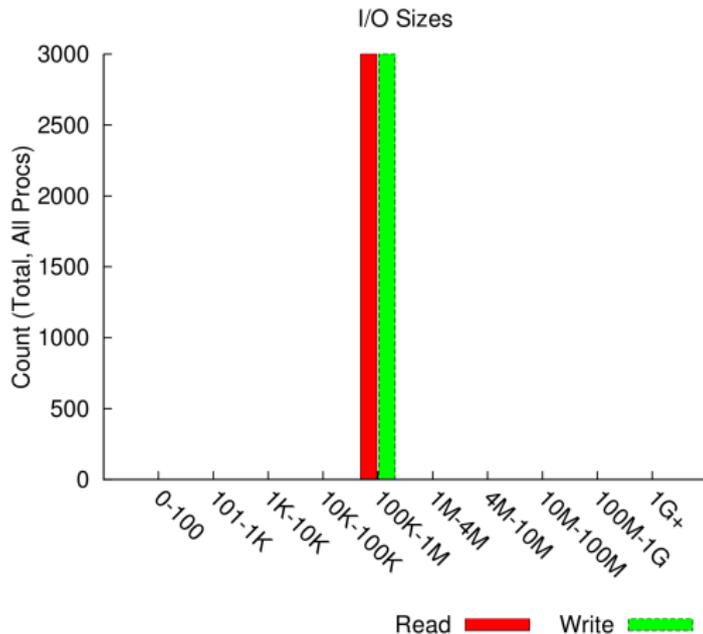
- Keine Berechnungen, trotzdem sehr hoher Other-Anteil

Darshan...



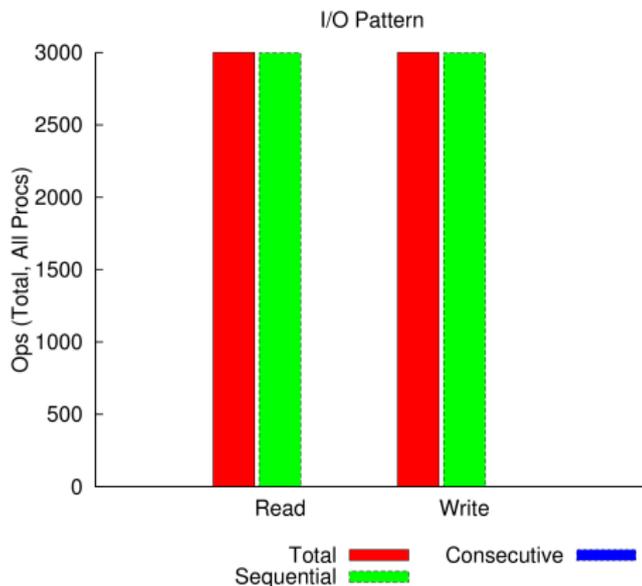
- Wie erwartet 3.000 Schreib- und Leseoperationen
 - $10 \text{ Prozesse} \times 100 \text{ Operationen} \times 3 \text{ Wiederholungen}$

Darshan...



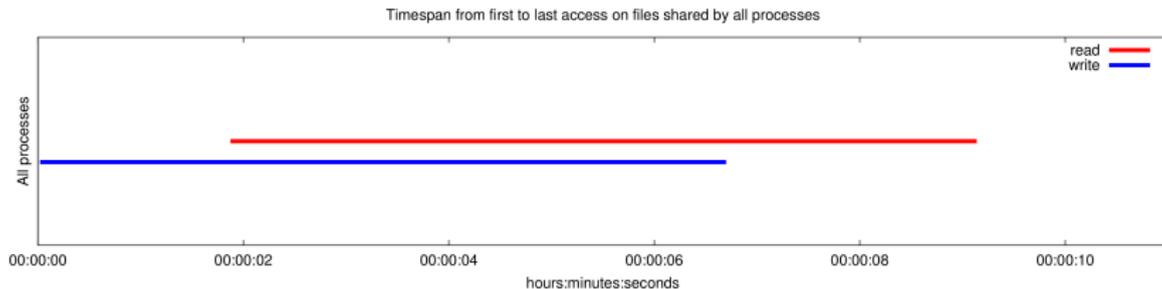
- Alle Operationen haben 1 MiB Größe

Darshan...



- Sequential: Zugriffe mit aufsteigendem Offset
- Consecutive: Zugriffe direkt hintereinander

Darshan...



- Anzeige durch drei Wiederholungen verräterisch
- Grobe Übersichten der Kosten von E/A
 - Bezüglich Aufrufanzahl, Zugriffsgröße und -muster
- Erlaubt Abschätzung ob Optimierung notwendig ist
 - Häufig genauere Analyse notwendig
 - Neuer Modus: Darshan eXtended Tracing (DxT)

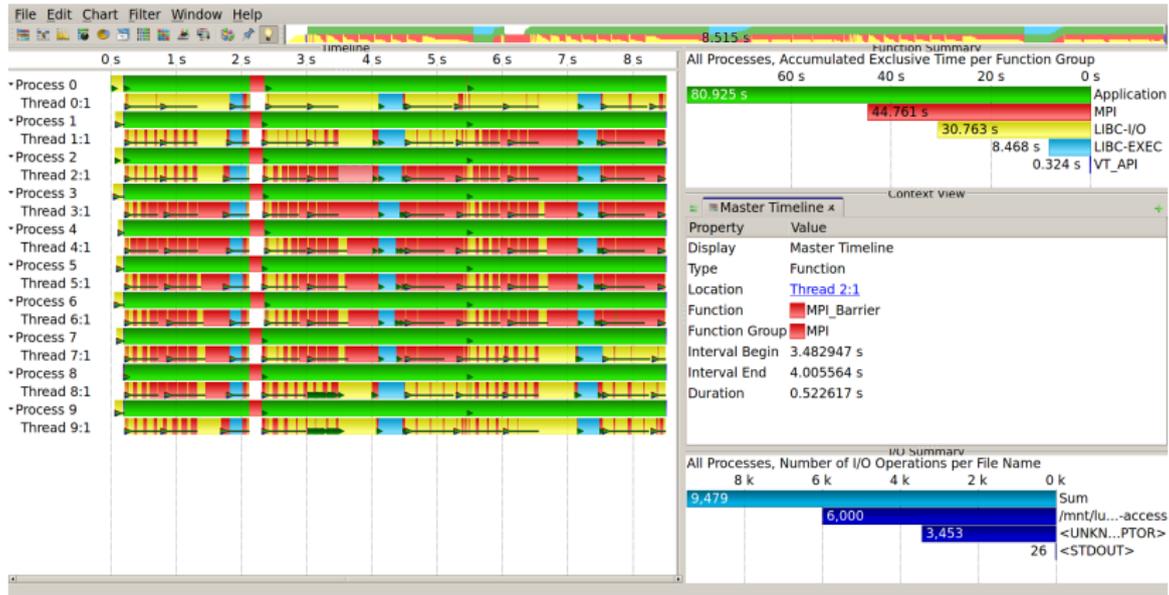
Score-P

- Score-P zeichnet Spurdaten auf
 - Score-P ist Open Source
 - Vorgänger: VampirTrace
 - Spezifisch für eine MPI-Implementierung
 - Compiler-Wrapper scorep
- Vampir zeigt Spurdaten an
 - Vampir ist kommerziell
 - Evaluationslizenzen verfügbar
- Spurdaten sind deutlich größer als Darshan-Protokolle
 - Im getesteten Fall mehr als Faktor 100

Score-P...

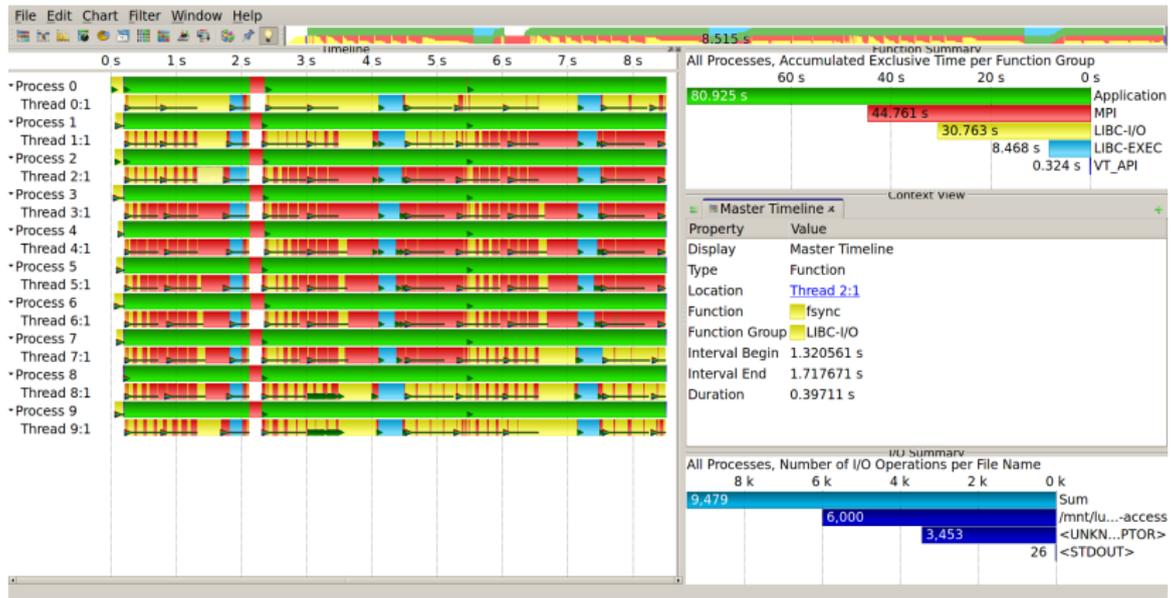
- Tracing erzeugt signifikanten Overhead
 - Laufzeit deutlich höher
 - Eventuell abweichendes Laufzeitverhalten
- Momentan noch keine E/A-Unterstützung
 - Dafür noch VampirTrace notwendig

Score-P...



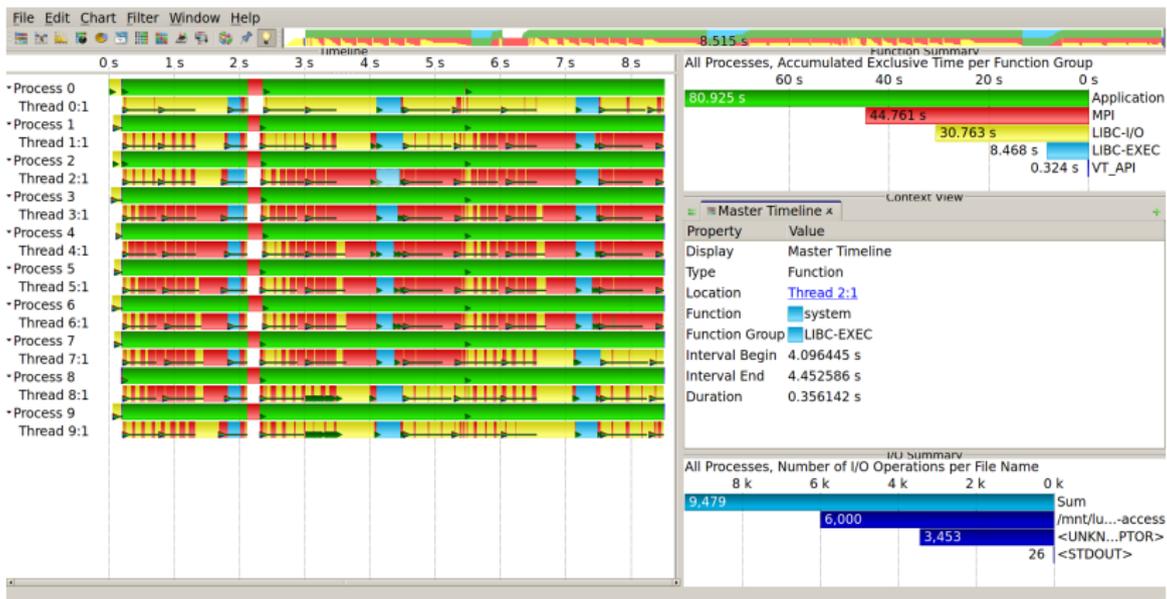
- Hauptthreads schlafen durchgängig
- Sehr ungleiche E/A-Zeiten, dadurch lange Barries

Score-P...



■ Synchronisieren dauert teilweise sehr lange

Score-P...



- Leeren des Caches auch sichtbar
 - Blockiert bis Blöcke geschrieben sind

Einleitung

- Bewertung der Leistung durch Modellierung der theoretisch möglichen Leistung
- Dazu sind einige Informationen notwendig
 - Involvierte Komponenten
 - Leistungscharakteristika der Komponenten
- Zusätzliche Leistungsmessungen der Komponenten
 - Dazu wieder andere Werkzeuge

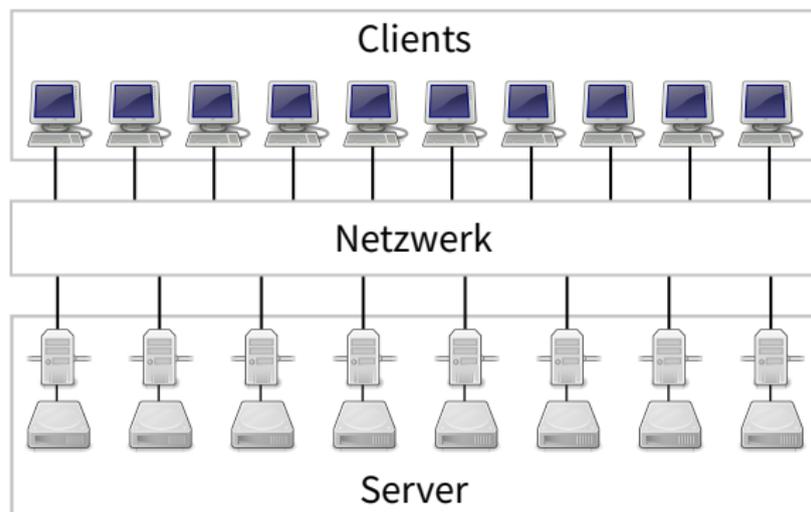
Beispiel

- Beispiel: Echtzeit-Twitter-Analyse
 - System kann in Echtzeit Stimmung in Tweets bestimmen
 - Alternativ Tweets speichern und später analysieren
- Frage: Wie ist die Leistung zu bewerten?

Beispiel

- Beispiel: Echtzeit-Twitter-Analyse
 - System kann in Echtzeit Stimmung in Tweets bestimmen
 - Alternativ Tweets speichern und später analysieren
- Frage: Wie ist die Leistung zu bewerten?
 - Ca. 6.000 Tweets pro Sekunde
 - 140 Bytes pro Tweet entspricht 840 KB/s
 - 26,5 TB pro Jahr

Übersicht



- Clients: IOPS, RAM, Bus zum Netzwerk
- Netzwerk: Durchsatz und Latenz
- Server: Durchsatz und IOPS der Speichergeräte, Bus

Client

- Anzahl der E/A-Operationen pro Sekunde
 - Kontextwechsel könnten Beschränkung sein
- Durchsatz und Latenz des Hauptspeichers
 - Üblicherweise kein Problem
- Abschätzung mit Hilfe von `tmpfs` und `fiio`
 - Idee: Viele kleine E/A-Operationen ausführen

```
1 $ mkdir /tmp/fs
2 $ mount -t tmpfs tmpfs /tmp/fs
3 $ ...
4 $ umount /tmp/fs
```


Speichergerät

- Deutliche Unterschiede je nach Speichertechnologie
 - HDD vs. SSD
- Durchsatz der Speichergeräte
 - Niedriger als Netzwerkdurchsatz für maximale Ausnutzung
 - Höher als Netzwerkdurchsatz für Leistungsreserven
- IOPS wichtig für Metadatenoperationen
 - Auch für zufällige Datenzugriffe
- Speicherbus kann auch beschränkender Faktor sein
 - Teilweise immer noch SATA 2.0 (300 MB/s)

Beispiel

- Messung des Datendurchsatzes
 - Lustre und OrangeFS
- Unterschiedliche Blockgrößen
 - 1 MiB und 64 KiB
 - Entspricht den Streifenbreiten von Lustre und OrangeFS
- Bei 9.090 Nachrichten pro Sekunde
 - 9,09 GiB/s (1 MiB) bzw. 582 MiB/s (64 KiB) pro Knoten
- Flaschenhals ist Netzwerkdurchsatz
 - Maximal 1.120 MiB/s

Beispiel...

Dateisystem	Blockgröße	1 PPN	6 PPN	12 PPN
Lustre	1 MiB	640 MiB/s	105 MiB/s	110 MiB/s
OrangeFS	1 MiB	160 MiB/s	390 MiB/s	430 MiB/s
OrangeFS	64 KiB	250 MiB/s	115 MiB/s	180 MiB/s

Tabelle: Schreiben

Dateisystem	Blockgröße	1 PPN	6 PPN	12 PPN
Lustre	1 MiB	1.095 MiB/s	735 MiB/s	875 MiB/s
OrangeFS	1 MiB	130 MiB/s	265 MiB/s	430 MiB/s
OrangeFS	64 KiB	505 MiB/s	140 MiB/s	195 MiB/s

Tabelle: Lesen

- Massiver Leistungseinbruch bei Lustre
 - Beim Schreiben kein exklusiver Zugriff auf OST
- Streifenbreite als Blockgröße teilweise vorteilhaft

Beispiel...

Dateisystem	Blockgröße	1 PPN	4 PPN	8 PPN
Lustre	$\frac{1}{PPN}$ MiB	640 MiB/s	620 MiB/s	605 MiB/s
OrangeFS	$\frac{64}{PPN}$ KiB	250 MiB/s	280 MiB/s	210 MiB/s

Tabella: Schreiben

Dateisystem	Blockgröße	1 PPN	4 PPN	8 PPN
Lustre	$\frac{1}{PPN}$ MiB	1.095 MiB/s	1.800 MiB/s	525 MiB/s
OrangeFS	$\frac{64}{PPN}$ KiB	505 MiB/s	655 MiB/s	455 MiB/s

Tabella: Lesen

- Bessere Leistung in beiden Fällen
- Anomalie beim Lesen von Lustre

Zusammenfassung

- Benchmarks erlauben die Erfassung der aktuellen Leistung
 - Viele unterschiedliche Benchmarks für viele unterschiedliche Anwendungsfälle
- Werkzeuge sind notwendig zur eingehenden Analyse
 - Sowohl für Grobüberblick als auch zur Detailanalyse
- Messung sagt nichts über mögliche Leistungsfähigkeit aus
 - Dafür ist eine Bewertung der Leistung notwendig
 - Häufig durch Leistungsmodellierung

Zusammenfassung...

- Grobe Modellierung ist häufig schon ausreichend
 - Lässt sich bei Bedarf verfeinern
- Unvorhersehbares Verhalten macht Analyse schwierig
 - Siehe beispielsweise Leseleistung bei Lustre
- Analyse der konkreten Implementierung notwendig
 - Optimierung setzt viele Detailkenntnisse voraus

- 1 Leistungsanalyse
 - Orientierung
 - Einleitung
 - Leistungsmessung
 - Leistungsbewertung
 - Zusammenfassung

2 Quellen

Quellen I

- [1] CHAOS Development Team. IOR - Parallel filesystem I/O benchmark. <https://github.com/chaos/ior>.
- [2] Jens Axboe. fio - Flexible IO Tester. <http://git.kernel.dk/?p=fio.git;a=summary>.
- [3] Hongzhang Shan and John Shalf. Using IOR to Analyze the I/O Performance for HPC Platforms. In *In: Cray User Group Conference (CUG'07, 2007)*.