

Hochleistungs- Eingabe/Ausgabe

- ▶ Motivation
- ▶ Abstraktionsebenen
- ▶ Traditionelle und moderne E/A
- ▶ E/A-Klassen bei numerischen Anwendungen
- ▶ Benutzungsschnittstellen
- ▶ Parallel Virtual File System (PVFS/PVFS2)
- ▶ Forschungsthemen (allgemein und hier)

Hochleistungs-Eingabe/Ausgabe

Die zehn wichtigsten Fragen

- ▶ Warum ist E/A auf einmal ein Thema?
- ▶ Wie hantiert man mit Daten in Dateien?
- ▶ Welche Abstraktionsschichten unterscheidet man?
- ▶ Welche Varianten der E/A finden wir bei numerischen Anwendungen?
- ▶ Welche Benutzungsschnittstellen gibt es?
- ▶ Wie funktioniert E/A im Cluster?
- ▶ Wie funktioniert E/A im Hochleistungsrechner?
- ▶ Was ist ein paralleles Dateisystem?
- ▶ Wie funktioniert das Parallel Virtual File System?
- ▶ Welche offenen Fragestellungen gibt es?

Warum ist das ein Thema?

- ▶ Gespeicherte Datenmengen steigen stark an
 - ▶ Berkeley-Bericht „How Much Information? 2003“
 - ▶ 2002: 5 Exabyte **mehr** abgespeichert
 - ▶ 92% davon auf magnetische Medien, meistens Festplatten
 - ▶ IDC-Report sieht für 2009 800 Exabyte gespeichert
 - ▶ Vermutet für 2010 einen Zuwachs von 400 Exabyte
- ▶ Hauptspeichergrößen steigen stark an
 - ▶ Heute einzelne Gigabyte pro Rechner
 - ▶ Z.B. DKRZ/Blizzard: 20 TByte Hauptspeicher, 6 Pbyte Platten
 - DKRZ/Mistral Faktor 10 größer
 - ▶ Füllt 5 bzw. 40 aktuelle große Festplatten
 - vgl. PC: 4 GByte Hauptspeicher vs. 4 TByte Platte
 - ▶ 1 TByte lesen bei 50 MByte/s dauert 20.000 s

Speicherung großer Datenmengen

▶ Datenaufkommen

- ▶ Besonders hoch in den Naturwissenschaften
Klimaforschung, Physik, Biologie, Astronomie, ...

▶ Zugriff

- ▶ Alle Anwender wollen immer alle Daten aufheben und sie jederzeit zugreifbar haben

▶ Verfügbarkeit

- ▶ Die Datenspeicherung soll mit Fehlern in der Hardware problemlos umgehen können

▶ Sicherheit

- ▶ Daten müssen vor Einblick, Veränderung und Löschen geschützt werden

Komplexere E/A-Systeme

- ▶ RAID – Redundant Array of Inexpensive Disks
- ▶ MAID – Massive Array of Idle Disks
- ▶ JBOD – Just a Bunch of Disks

- ▶ SAN – Storage Area Network (blockorientiert)
- ▶ NAS – Network Attached Storage (dateiorientiert)

- ▶ Dateisysteme mit verteiltem Zugriff
 - ▶ NFS – Network File System
 - ▶ AFS – Andrew File System
- ▶ Dedizierte Spezial-Hardware in Hochleistungsrechnern

Abstraktionsebenen

Begriff der „parallelen Eingabe/Ausgabe“

- ▶ Programmsicht:

Der Zugriff auf die Bytes *einer* Datei erfolgt aus mehreren parallelen Prozesse heraus

- ▶ Systemsicht:

Die Bytes einer Datei liegen über mehrere Platten verteilt

Wie üblich: Ebenen voneinander unabhängig

Varianten der E/A

▶ Traditionell

- ▶ E/A nur über Hostrechner
- ▶ E/A nur durch festgelegten Prozess

Bewertung

- ▶ Aus Effizienzgründen nicht mehr möglich

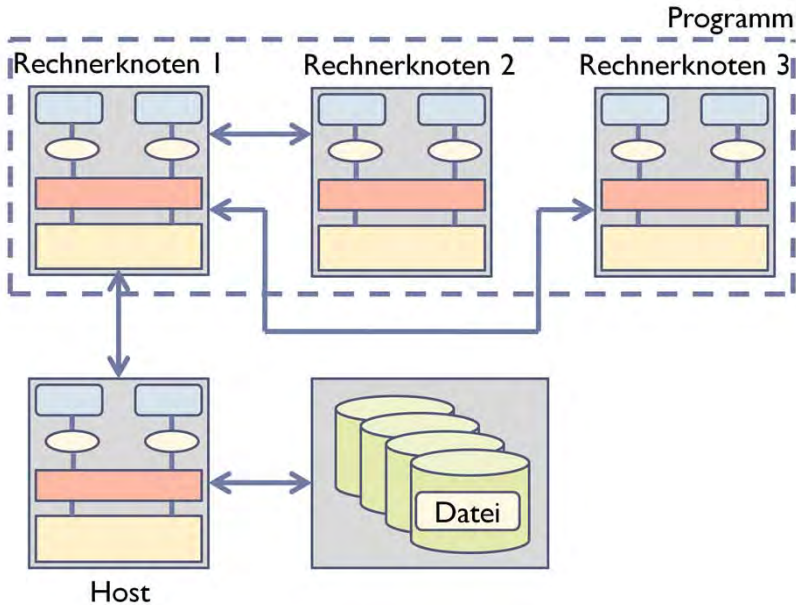
▶ Modern

- ▶ E/A über viele Knoten
- ▶ E/A durch alle Prozesse

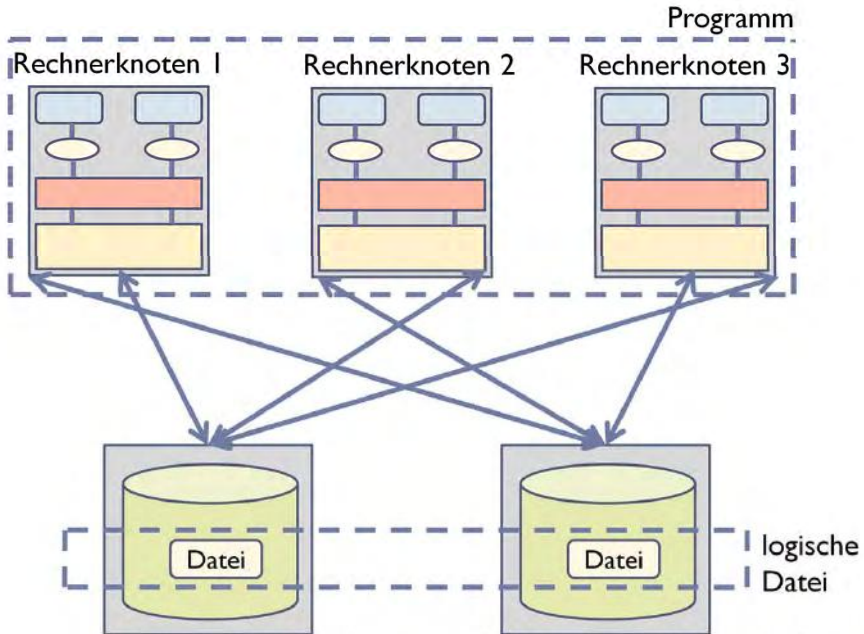
Bewertung

- ▶ Verwendung moderner Techniken

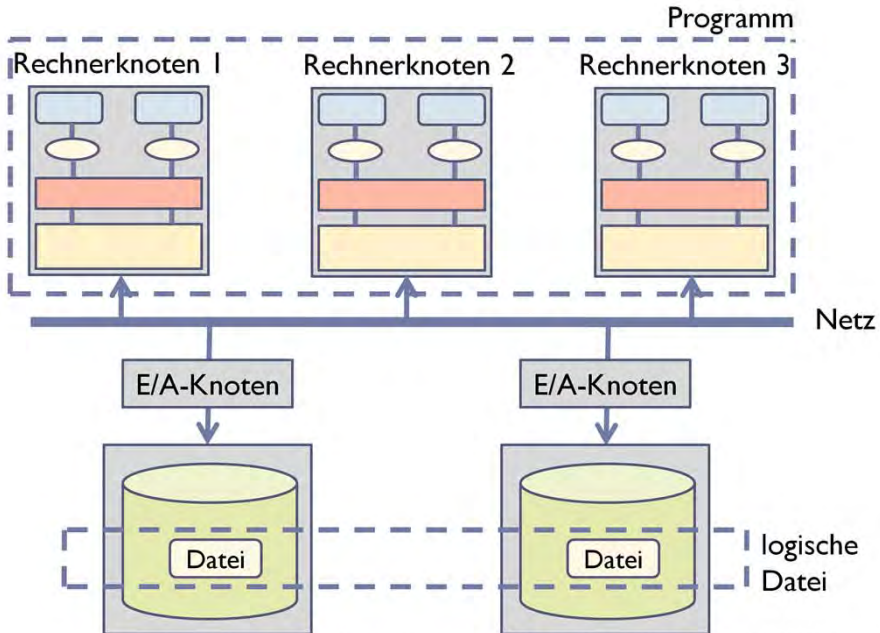
Traditionelle E/A



Moderne E/A



Moderne E/A...



Geschichte der parallelen E/A

- ▶ Parallele E/A-Systeme und parallele E/A-Bibliotheken entstehen Anfang der 90er Jahre
- ▶ Viel Forschung bereits Mitte der 90er
- ▶ Nur wenige existierende Systeme im Produktionsbetrieb
- ▶ Aber: Sehr viele proprietäre Hochleistungs-E/A-Systeme bei Hochleistungsrechnern; jedoch nicht so oft echt parallele

Kategorien massiver E/A

Anwendungssicht

- ▶ Nichtnumerische Anwendungen
 - ▶ Unregelmäßig strukturierte Daten
z.B. Datenbank-Anwendungen
 - ▶ Datenströme
z.B. Multimedia-AnwendungenBeides hier nicht weiter betrachtet
- ▶ Numerische Anwendungen
 - ▶ Regelmäßig strukturierte Daten
z.B. Vektoren, Arrays mit großen Dimensionen
 - ▶ Auch unregelmäßig strukturierte Daten
 - Listen, dünnbesetzte Matrizen

E/A-Klassen bei numerischen Anwendungen

- ▶ Lesen der Programmeingabe und Schreiben der Programmausgabe
- ▶ Sicherungspunkte
- ▶ Temporäre Daten
- ▶ „Out-of-core Execution“

Programmeingabe/-ausgabe

- ▶ Zeitpunkte
 - ▶ Programmstart, Programmende, Zwischenergebnisse
- ▶ Datenmengen
 - ▶ Maximal Größe des gesamten Hauptspeichers
- ▶ Wichtiges Szenarium: Pipelining
 - ▶ Daten von einem anderen Gerät
z.B. von physikalischem Experiment
 - ▶ Daten zu einem anderen Gerät
z.B. Ergebnisvisualisierung, Datenarchivierung

Sicherungspunkte

- ▶ Verwendet zur Programmfortsetzung
 - ▶ Nach Absturz oder Unterbrechung
- ▶ Sichern aller wichtigen Daten
 - ▶ Kompletter Speicherabzug
 - ▶ Vom Benutzer ausgewählte Daten
- ▶ Anzahl benötigter Sicherungspunkte
 - ▶ Mindestens zwei
- ▶ Redundanz der Daten notwendig zur Ausfallsicherung

Temporäre Dateien

- ▶ Abspeicherung während des Programmlaufs
- ▶ Evtl. nicht-parallele E/A auf lokaler Platte ausreichend
- ▶ Zur Kommunikation zwischen Prozessen aber parallele E/A erforderlich
 - ▶ Verwendung *einer* temporären Datei über alle Platten hinweg

Out-of-Core-Execution

- ▶ **Out-of-core-Execution:**
Bearbeitung einer größeren Datenmenge, als in den Hauptspeicher paßt
 - ▶ Eigenprogrammiertes Aus- und Einlagern der überschüssigen Datenmengen
 - ▶ Effizienter als Swapping durch Betriebssysteme
- ▶ **Spezialfall für Spezialanwendungen**
 - ▶ Bei numerischen Anwendungen wird typischerweise der Hauptspeicher exakt gefüllt

Zugriffsmuster bei E/A

- ▶ Wichtig für Fall 1: E/A bei Programmen
- ▶ Fragestellung
 - ▶ Wann wird E/A durchgeführt?
 - ▶ Welche Mengen werden transferiert?
 - ▶ Welche Byte einer Datei werden angesprochen?
- ▶ Ausführliche Studien aus den 90ern
- ▶ Wir müssen hier lernen, was die Klimaforscher tun
- ▶ Wichtig um die Abbildung der logischen Datei auf die physikalische zu optimieren

Benutzungsschnittstellen

► Hierarchie von Schnittstellen

Im Moment nur paralleles Dateisystem betrachtet

(P)netCDF, HDF5	strukturierte Daten mit Annotationen
MPI-IO	Listen typisierter Daten
API paralleles Dateisystem	Listen typisierter Daten
POSIX open(), read(), write(), close(), usw.	Byte-Sequenz

E/A im Rechnercluster

- ▶ Gelegentlich: An jedem Knoten auch eine Platte
 - ▶ Entweder nur für temporäre Dateien
 - ▶ Oder als richtiges paralleles Dateisystem über alle Platten hinweg
- ▶ Alternativen
 - ▶ Jeder Rechenknoten ist auch E/A-Knoten
 - ▶ Dedizierte E/A-Knoten
 - Balancierung der Rechen- und E/A-Leistung
- ▶ Betriebsproblematik
 - ▶ Wer darf wann welche Platten benutzen?
(Bisher nur Konzepte für Knotenzuteilung)

E/A im Hochleistungsrechner

- ▶ Knoten haben nie Platten
- ▶ Ausgewählte Knoten dienen als E/A-Knoten
 - ▶ Dicker Netzanschluss
 - ▶ Fetter Hauptspeicherausbau
 - ▶ Keine Programme auf diesen Knoten
 - ▶ Keine eigenen Festplatten
- ▶ Wie geht es?
 - ▶ E/A-Knoten leitet die gesamte E/A zum E/A-System bestehend aus eigenen Rechnern und sehr vielen Platten

Paralleles Dateisystem

▶ Merkmale

- ▶ Mehrere Prozesse können gleichzeitig auf dieselben Dateien zugreifen.
- ▶ Die Daten einer Datei liegen physikalisch verteilt

▶ Schicht

- ▶ Zwischen dem Anwenderprogramm und dem physikalischen E/A-System der E/A-Knoten
- ▶ Somit typische Middleware-Software

- ▶ **IBM Spectrum Scale**

- ▶ Bis Feb. 2015 genannt GPFS (Cluster-Dateisystem)
- ▶ Langjähriges Produkt; ausgereiftes System

- ▶ **Lustre (Cluster-Dateisystem)**

- ▶ Neuerer Ansatz, in dem alles besser gemacht wird
- ▶ Sehr hohe Komplexität!
- ▶ Open-Source und frei erhältlich; viele Varianten

- ▶ **OrangeFS (Paralleles Dateisystem)**

- ▶ Bis 2007 entwickelt als PVFS/PVFS2
- ▶ Verbreitetes System bei Selbstbau-Clustern

- ▶ **BeeGFS**

- ▶ Entwickelt bei Fraunhofer Institut

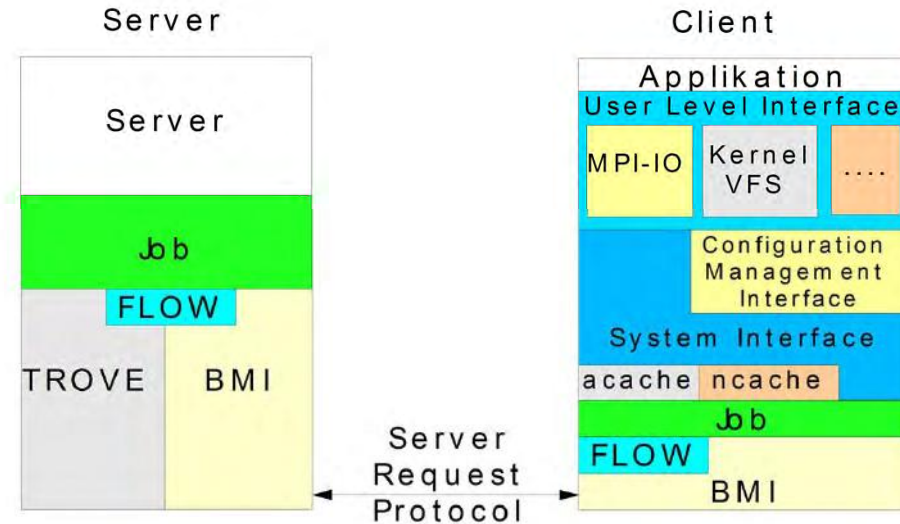
Beispiel: Parallel Virtual File System

- ▶ Ziel: E/A massiver Datenmengen in Clustern
- ▶ Entwicklergruppen
 - ▶ Parallel Architecture Research Laboratory
Clemson University
 - ▶ Mathematics and Computer Science Division
Argonne National Laboratories
- ▶ Historie
 - ▶ Beginn ~1996
 - ▶ PVFS2 (Herbst 2003) umbenannt in PVFS 2.0

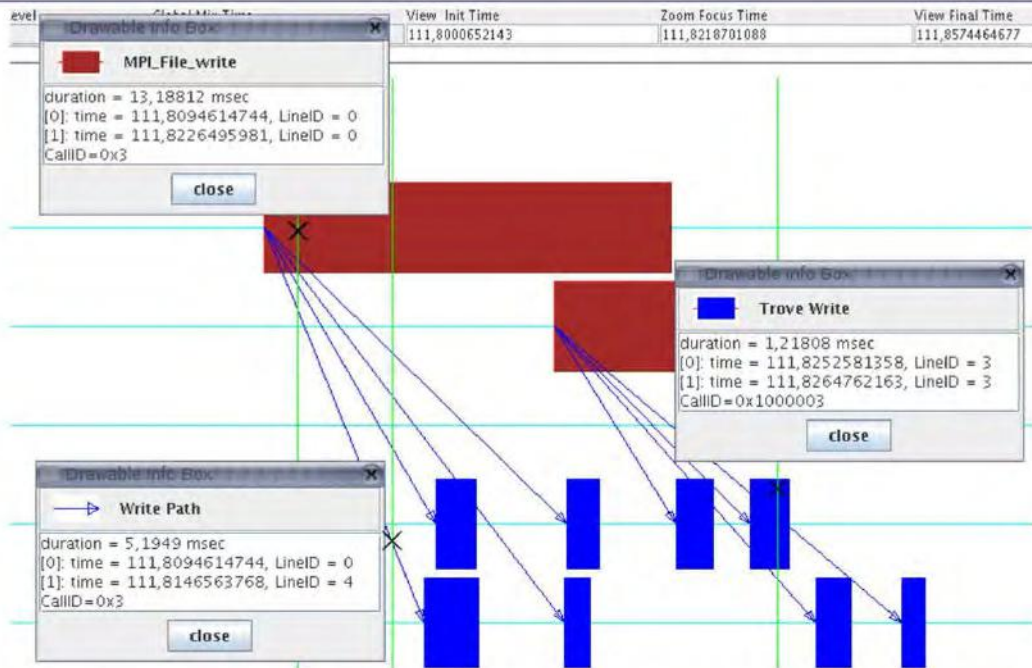
PVFS2-Eigenschaften

- ▶ Komplette Neuentwicklung (ggü. PVFS)
- ▶ Setzt auf realem Dateisystem auf (ext[23] o.ä.)
- ▶ Modular und hierarchisch aufgebaut
 - Module zum Auswechseln vorgesehen
- ▶ Enge MPI-IO-Integration
- ▶ Effiziente Durchführung strukturierter nicht-kontinuierlicher Zugriffe
- ▶ Zustandsloser Objektzugriff
- ▶ Erweiterbare Datenverteilungsfunktion (striping usw.)
- ▶ Semantik des Dateisystems variabel
- ▶ Explizite Unterstützung paralleler Abläufe
- ▶ Redundantes Speichern von Daten und Metadaten

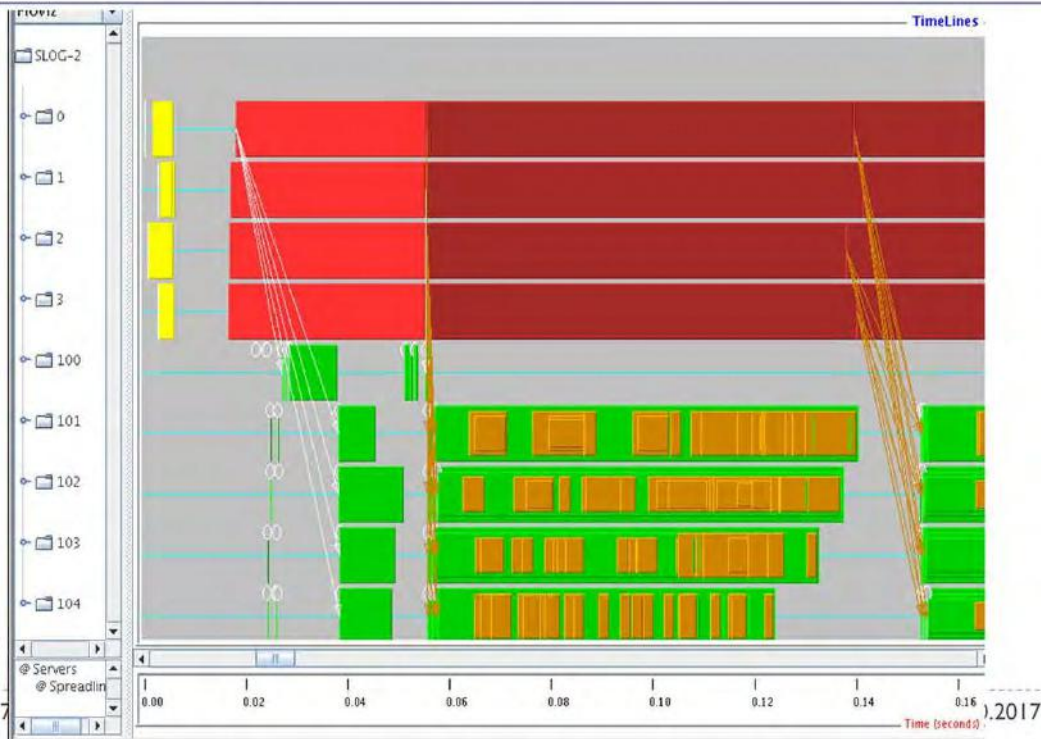
PVFS2-Schichtenmodell



MPI_File_write löst PVFS trove aus



4 Clients, 4 Server, 1 Metadatenserver



Arbeitsbereich Wissenschaftliches Rechnen und parallele E/A-Systeme

Der Arbeitsbereich Wissenschaftliches Rechnen (ehemals Arbeitsgruppe „Parallele und verteilte Systeme“ an der Universität Heidelberg) verwendet Lustre als Basis für eigene Forschungs- und Entwicklungsarbeiten auf dem Gebiet der parallelen E/A

Themen für Abschlußarbeiten und zur Mitarbeit sind vorhanden!

Forschungsthemen

- ▶ Wie soll parallele E/A genutzt werden?
- ▶ Wie steigern wir Leistung und Skalierbarkeit?
- ▶ Wie ermittelt man die Leistung des E/A-Systems?

Jetzt neu:

- ▶ Kombination mit Bandarchiv und Hierarchical Storage Management (HSM)
- ▶ Aspekte der Energie- und Kosteneffizienz

Forschungsthema Nutzung

- ▶ Die Frage der Schnittstelle zum Programm ist noch offen
 - ▶ Welche Zugriffsvarianten (Semantiken)?
 - ▶ Schnittstellen auf welchem Abstraktionsniveau?
- ▶ Details im Vortrag zu MPI-IO

Forschungsthema Leistung

- ▶ Zugriffsmuster erkennen
- ▶ Abbildung logische Daten auf physikalische Daten optimieren oder dynamisch gestalten
- ▶ Nichtzusammenhängende Zugriffe optimieren
- ▶ Kollektive Operationen unterstützen
- ▶ Metadatenzugriff verbessern
- ▶ Kompression einbauen

Allgemein: Skalierbarkeit steigern

Forschungsthema Leistungsbestimmung

- ▶ Keine standardisierten Benchmarks
- ▶ Was wollen wir messen?
 - ▶ E/A-Bandbreiten beim Datenzugriff
 - ▶ Verschiedene Zugriffsmuster
 - ▶ Verschiedene Datenmengen
 - ▶ Unabhängige/identische Orte in Dateien
 - ▶ Zugriffsraten beim Metadatenzugriff
- ▶ Zur Zeit keine vernünftigen quantitativen Vergleiche zwischen Systemen möglich

Eigene Forschungen

- ▶ Leistungsvervisualisierung
- ▶ Leistungsbewertung
- ▶ Modellierung und Simulation
- ▶ Metadatenverwaltung
- ▶ Anwendung im Produktionsbetrieb
- ▶ Zugriffsmuster
- ▶ Datenkomprimierung

Mitarbeit ist willkommen!

Hochleistungs-Eingabe/Ausgabe

Zusammenfassung

- ▶ Traditionelle Varianten der E/A jetzt ungenügend
- ▶ Parallelisierung der E/A notwendig und möglich
- ▶ Parallele E/A etwa ab Mitte der 90er entwickelt
- ▶ Uns interessieren hier nur numerische Anwendungen
- ▶ E/A gliedert sich in verschiedene Klassen auf
- ▶ Benutzerschnittstellen auf verschiedenen Ebenen
- ▶ Im Parallelen Dateisystem liegen die Dateien über Platten verteilt und werden von parallelen Prozessen aus gelesen und geschrieben
- ▶ Im Einsatz in der TOP500: Lustre und IBM
- ▶ Im Bereich E/A viele offene Forschungsfragen