Parallele 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)

Parallele 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
- Hauptspeichergrößen steigen stark an
  - Heute einzelne Gigabytes pro Rechner
    - Z.B. DKRZ/Blizzard: 20 TByte Hauptspeicher
    - Füllt 20 aktuelle große Festplatten

      vgl. PC: 4 GByte Hauptspeicher vs. 1 TByte Platte
    - 1 TByte lesen bei 50 MByte/s dauert 20.000 s

Siehe: http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/

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
- NAS – Network Attached Storage

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

Siehe: http://en.wikipedia.org/wiki/RAID , http://en.wikipedia.org/wiki/MAID , http://en.wikipedia.org/wiki/JBOD#JBOD

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

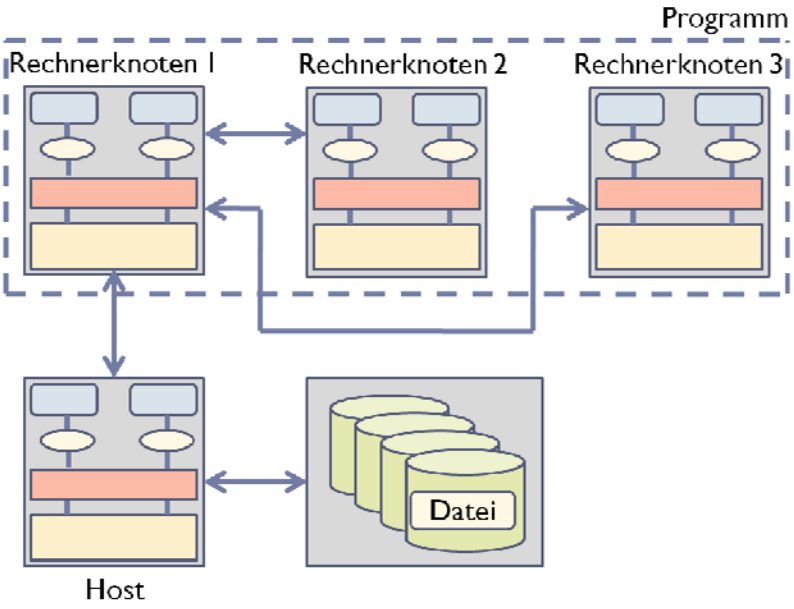
- Traditionell
  - E/A nur über Hostrechner
  - E/A nur durch festgelegten Prozeß

  Bewertung
  - Aus Effizienzgründen nicht mehr möglich

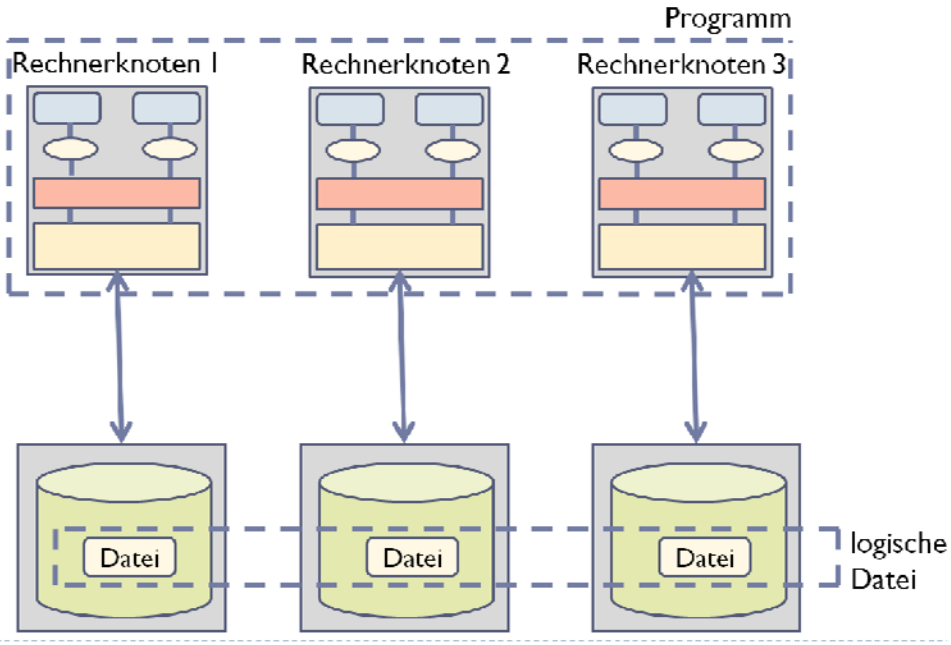
- Modern
  - E/A über viele Knoten
  - E/A durch alle Prozesse

  Bewertung
  - Verwendung moderner Techniken

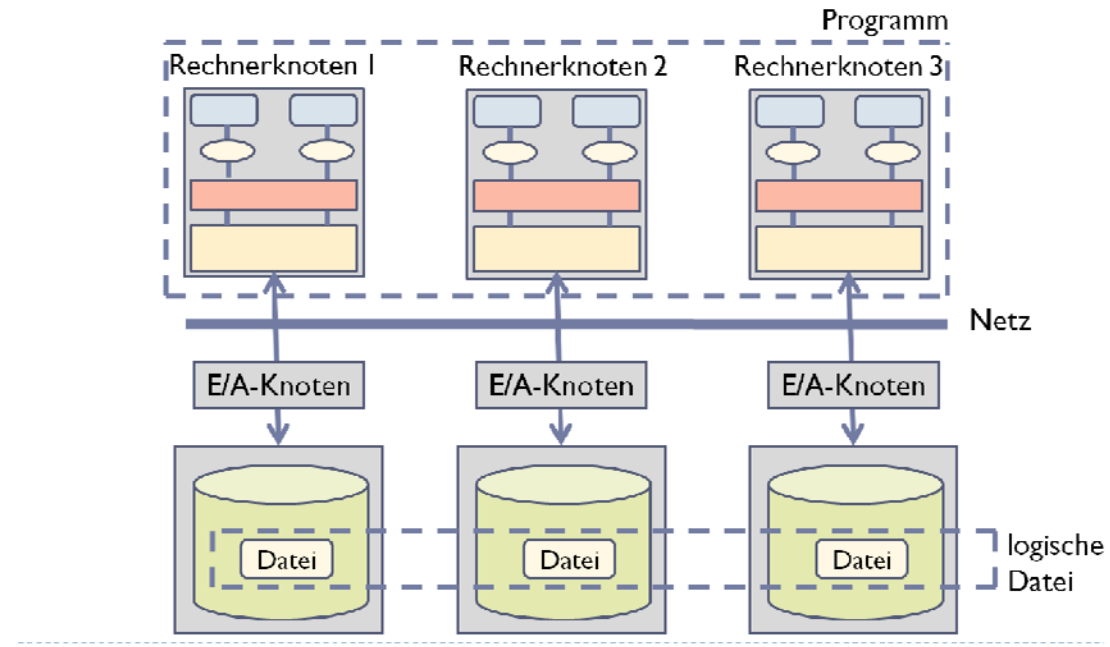
Traditionelle E/A



Moderne E/A


Programm
Rechnerknoten 1
Rechnerknoten 2
Rechnerknoten 3
Datei
Datei
Datei
logische
Datei

Moderne E/A...


Programm
Rechnerknoten 1
Rechnerknoten 2
Rechnerknoten 3
Netz
E/A-Knoten
E/A-Knoten
E/A-Knoten
Datei
Datei
Datei
logische
Datei

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 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

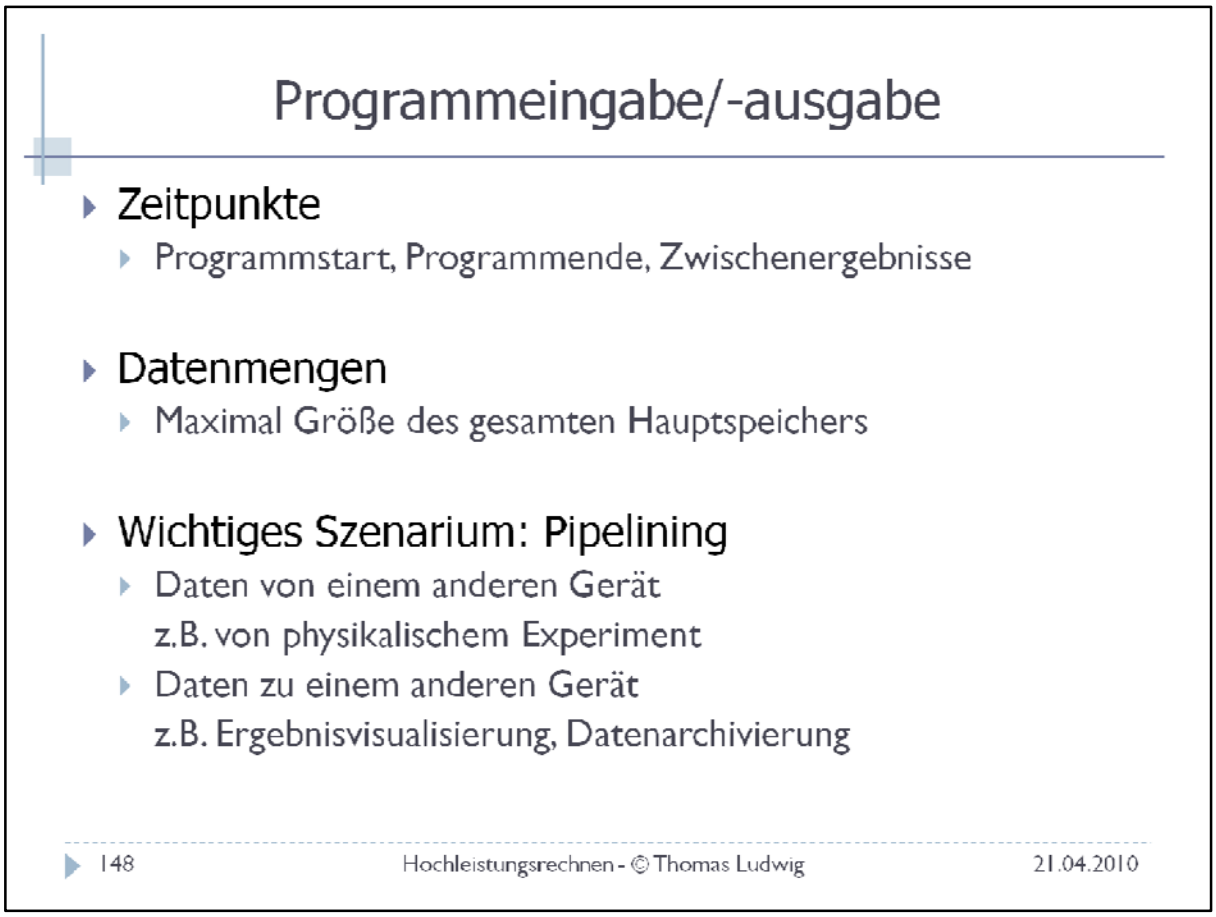
    z.B. Multimedia-Anwendungen

  Beides hier nicht weiter betrachtet
- Numerische Anwendungen
  - Regelmäßig strukturierte Daten

    z.B. Vektoren, Arrays mit großen Dimensionen

E/A-Klassen bei numerischen Anwendungen

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



Pipelining-Modus mit massiven Datenmengen ist ein wichtiger künftiger Anwendungsfall.

Sicherungspunkte

- Sichern aller wichtigen Daten
- Verwendet zur Programmfortsetzung
  - Nach Absturz oder Unterbrechung
- 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 Betriebssystems

- Spezialfall für Spezialanwendungen
  - Bei numerischen Anwendungen wird typischerweise der Hauptspeicher exakt gefüllt

Zugriffsmuster bei E/A

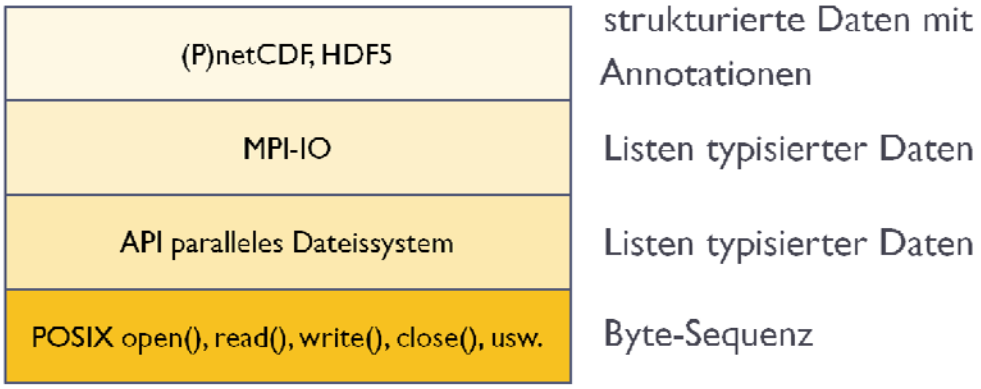
- Wichtig für Fall 1: E/A bei Programmen
- Fragestellung
  - Wann wird E/A durchgeführt?
  - Welche Mengen werden transferiert?
  - Welche Byteseiner Datei werden angesprochen?
- Ausführliche Studien aus den 90ern
- Wir müssen hier lernen, was die Klimaforscher tun
- Wichtig um die Abbildung der logischen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

Entspricht der Fragestellung der Datenaufteilung bei parallelen Programmen und Rechnern mit verteiltem Speicher: Welche Daten sollen wo liegen?

Benutzungsschnittstellen

- Hierarchie von Schnittstellen

  Im Moment nur paralleles Dateisystem betrachtet



Auf den unterschiedlichen Ebenen werden Objekte verschiedener Abstraktion übertragen. Ganz unten nur einzelne Byte. In den Schichten darüber schreibt/liest man mit einem Aufruf ganze Listen typisierter Daten. Ganz oben gleich komplexe Daten samt ihrer Annotationen.

E/A im Rechnercluster

- Häufig: 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

  Balanzierung 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

Ein paralleles Dateisystem stellt somit für die Hintergrunddaten so etwas ähnliches dar, wie ein virtuell gemeinsamer Speicher für die Hauptspeicherdaten.

Systeme

- GPFS (Cluster-Dateisystem)
  - Produkt der IBM; ausgreiftes System
- PVFS/PVFS2 (Paralleles Dateisystem)
  - Verbreitetes System bei Selbstbau-Clustern
- Lustre (Cluster-Dateisystem)
  - Neuerer Ansatz, in dem alles besser gemacht wird
  - Sehr hohe Komplexität!
  - Open-Source und frei erhältlich

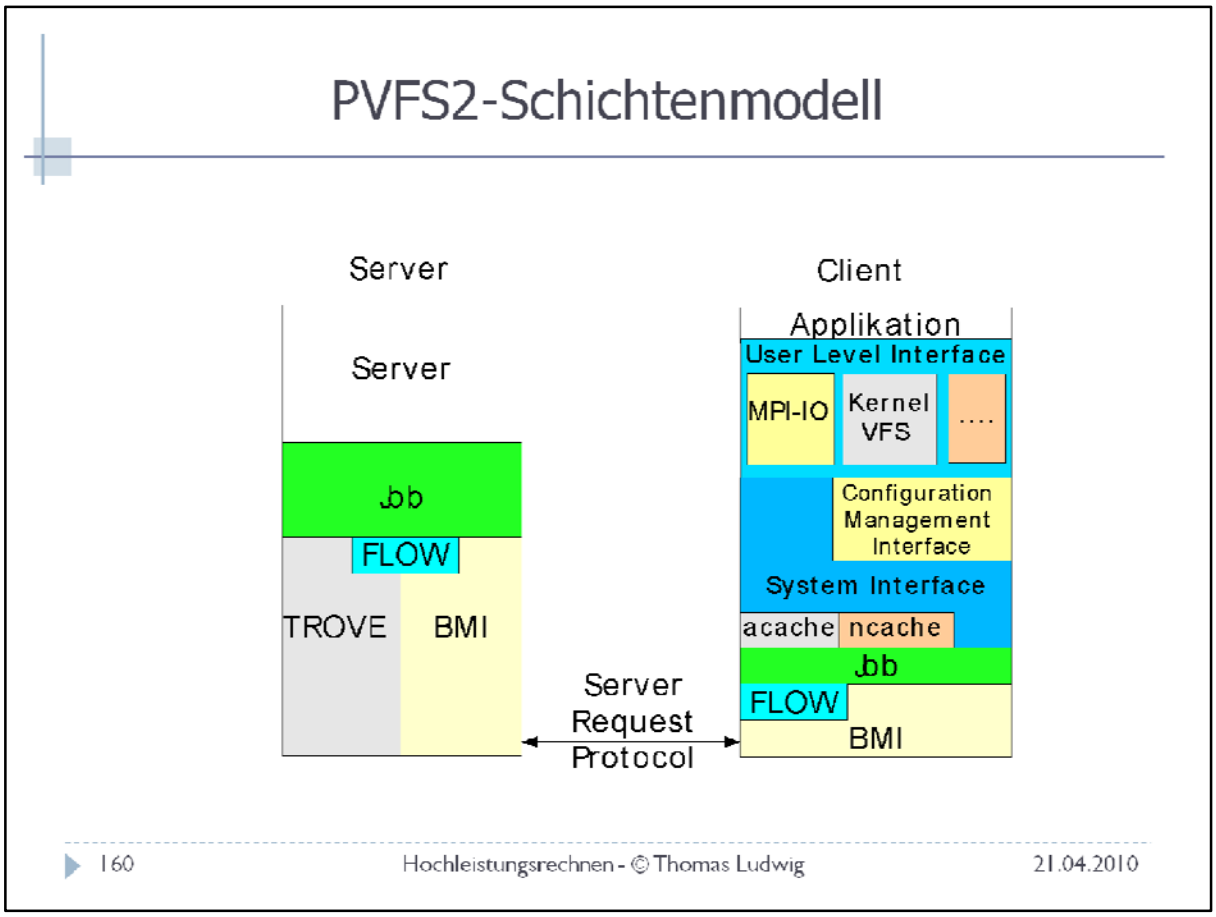
Beispiel: Parallel Virtual File System

- 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

Siehe: http://www.pvfs.org/

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
- Zustandloser Objektzugriff
- Erweiterbare Datenverteilungsfunktion (striping usw.)
- Semantik des Dateisystems variabel
- Explizite Unterstützung paralleler Abläufe
- Redundantes Speichern von Daten und Metadaten



Die Schicht Job kontrolliert die Abarbeitung einzelner Anfragen in all ihren Phasen. Es sind immer mehrere Schritte in den darunterliegenden Schichten auszuführen.

Flow: Steuert den Floß von Anfragen und Antworten zwischen den tieferen Schichten Trove und BMI.

BMI (Buffered Message Interface): Netzwerk-Abstraktionsschicht. Ermöglicht die Verwendung verschiedener Geräte und Protokolle. Zur Zeit für TCP/IP (Ethernet), Myrinet´s GM und Infiniband.

Trove: Abtraktionsschicht für die persistente Datenspeicherung.

Namens-Cache (ncache) und Attribute-Cache (acache).

Arbeitsbereich Wissenschaftliches Rechnen und PVFS

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

Themen für Abschlußarbeiten und zur Mitarbeit sind genügend vorhanden!

Forschungsthemen

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

Jetzt neu:

- Umstieg auf GPFS
- Kombination mit Bandarchiv und Hierarchical Storage Management (HSM)

Als HSM-System verwendet das DKRZ HPSS (High Performance Storage System)

Siehe: http://www.hpss-collaboration.org/

Forschungsthema Nutzung

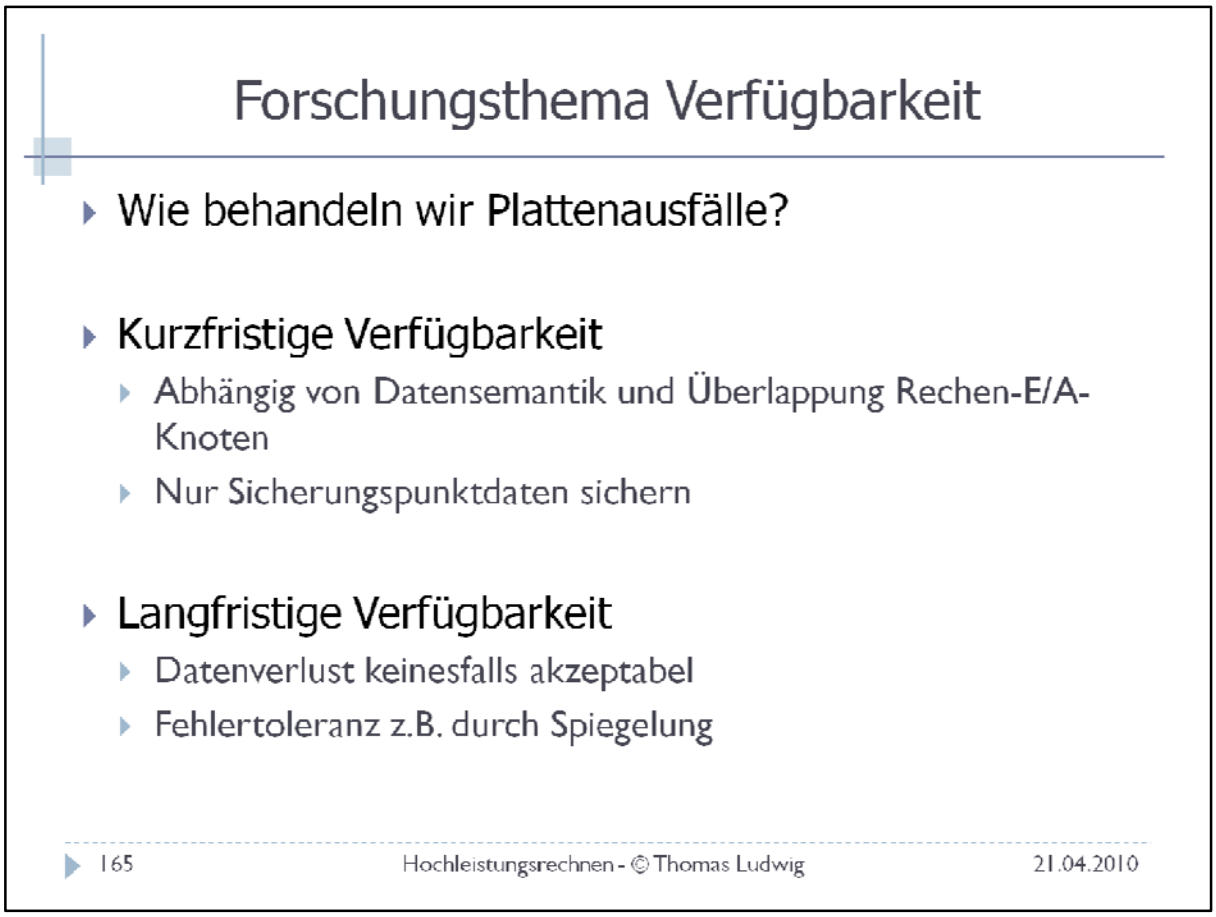
- Die Frage der Schnittstelle zum Programm ist noch offen
  - Welche Zugriffsvarianten (Semantiken)?
  - Schnittstellen auf welchem Abstraktionsniveau?
- Details im Vortrag zu MPI-IO

Forschungsthema Leistung

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

Allgemein: Skalierbarkeit steigern

Kollektive Operationen werden bei MPI-IO besprochen.



Bei Datensemantik „Sicherungspunkt“ müssen wir Fehlertoleranz einbauen, wohingegen bei „temporäre Daten“ dies nicht notwendig ist.

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

- Leistungsvisualisierung
- Leistungsbewertung
- Modellierung und Simulation
- Metadatenverwaltung
- Anwendung im Produktionsbetrieb
- Zugriffsmuster

Mitarbeit ist willkommen!

Parallele 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
- PVFS ist prominenter Vertreter dieser E/A-Systeme
- Im Bereich E/A viele offene Forschungsfragen