Hardware-Architekturen

1. Parallelismus
2. Klassifikation nach Flynn
3. Gemeinsamer und verteilter Speicher
4. Skalierbarkeit
5. Verbindungsnetze und Topologien
6. Hintergrundspeicher
7. Spezialkonzepte

1. Parallelismus

Unter Parallelverarbeitung verstehen wir die parallele Verwendung (gleichartiger) Komponenten zur Erzielung höherer Leistung

Beispiel

- Ein Arbeiter schaufelt ein 1m*1m*1m großes Loch in einer Stunde
- Zwei Arbeiter schaufeln ein 1m*1m*1m großes Loch in einer halben Stunde
- Tausend Arbeiter? ☹
- Ein 1000m*1m*1m großes Loch? ☺

Ebenen des Parallelismus im Rechner

- Parallele Rechnerarchitekturen
  - Besitzen Verarbeitungseinheiten, die koordiniert gleichzeitig an einer Aufgabe arbeiten
- Verarbeitungseinheiten
  - (Auch die Bits in einem Byte/Wort)
  - Spezialisierte Einheiten wie z.B. Rechenwerke
  - Prozessorkerne
  - Prozessoren
  - Vollständige Rechner
  - Hochleistungsrechnersysteme

Ebenen des Parallelismus (2)

- Vernetzung der Rechner
  - Viele parallele Wege zwischen zwei Verbindungspunkten
- Datenspeicherung
  - Viele Festplatten zu einem Verbund geschaltet
  - Viele Bandlesegeräte zu einem Verbund geschaltet

Ebenen des Parallelismus (3)

Prozessortechnologie bis ca. 2005

- Erhöhung der Frequenz → höhere Leistung
  - Entspricht quasi stärkerem Bauarbeiter
- Zusätzliche Prozessoren → noch mehr Leistung

Prozessortechnologie ab ca. 2005

- Frequenzerhöhung nicht weiter möglich, da Abwärme zu hoch – weiter Miniaturisierung klappt allerdings
  - Deshalb: mehrere vollständige Teilprozessoren (Kerne) in einem Prozessor

Ebenen des Parallelismus (4)

Prozessortechnologie der 80er Jahre

- Thinking Machines Corporation produziert die Connection Machine
- CM-1 (1985): 65.536 Ein-Bit-Prozessoren

Seymour Cray (1925-1996)

If you were plowing a field, what would you rather use? Two strong oxen or 1024 chickens?

W. Gropp, E. Lusk, A. Skjellum

To pull a bigger wagon, it is easier to add more oxen than to grow a giant ox.

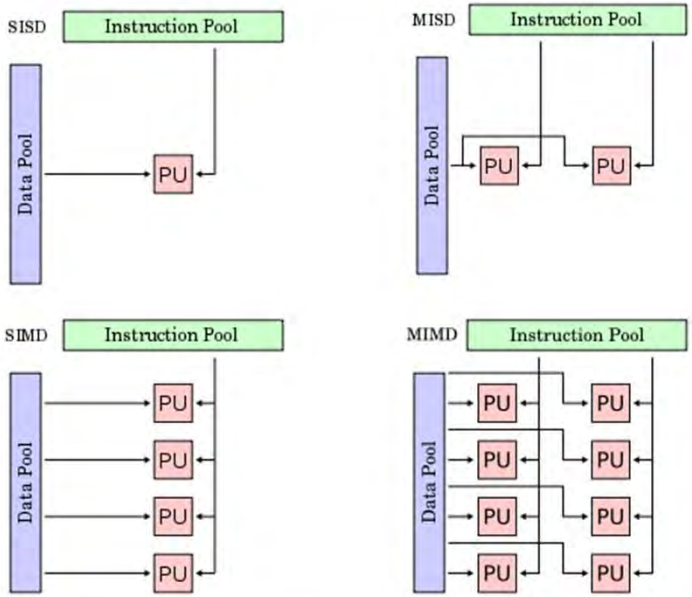
2. Klassifikation nach Flynn

Klassifikation nach Flynn (1972)

- Rechner arbeiten mit Befehlsströmen und Datenströmen
- Aus ihrer Kombination ergeben sich 4 Varianten

- SISD single instruction, single data stream
- SIMD single instruction, multiple data stream
- MISD multiple instruction, single data stream
- MIMD multiple instruction, multiple data stream

Klassifikation nach Flynn (2)



Klassifikation nach Flynn (3)

Was ist was bei Flynn?

- SISD: klassische von-Neumann-Architektur Monoprozessor-Rechner
  - quasi ausgestorben
- SIMD: Vektorrechner und Feldrechner
  - umgesetzt in Spezialarchitekturen wie z.B. Grafikkarten
- MISD: diese Klasse ist leer
- MIMD: alles, was uns interessiert
  - die Mehrprozessorsysteme

Unterteilung von Flynns MIMD-Klasse

Die Rechner bestehen aus mehreren Prozessoren, die über ein Verbindungsnetz kommunizieren

- Über die Verbindungen erfolgt der Informationsaustausch zwischen Prozessen auf verschiedenen Prozessoren sowie Synchronisation und Kooperation

Neue Unterscheidungskriterien

- Wie sehen die Prozessoren den Adressraum des Speichers?
- Wie sind die Speicherkomponenten mit dem Prozessor gekoppelt?

3. Gemeinsamer & verteilter Speicher

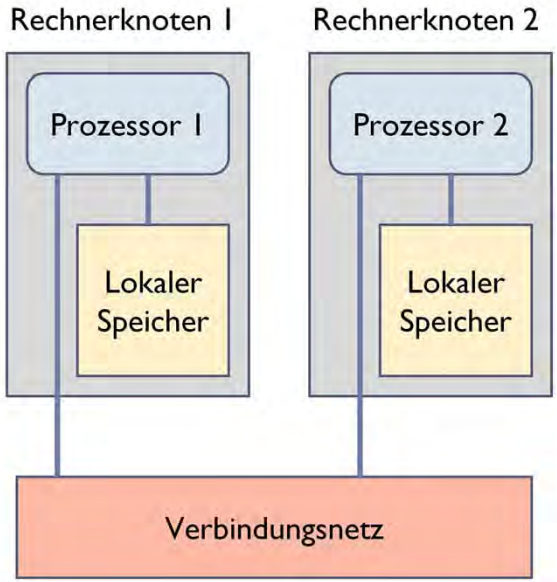
Neue Klassen von Rechnerarchitekturen

- Rechner mit verteiltem Speicher
- Rechner mit gemeinsamem Speicher
- Mischformen

Mischformen sind in der Informatik sehr beliebt

- Man versucht, die Vorteile der Ansätze zu vereinen, ohne die Nachteile in Kauf nehmen zu müssen

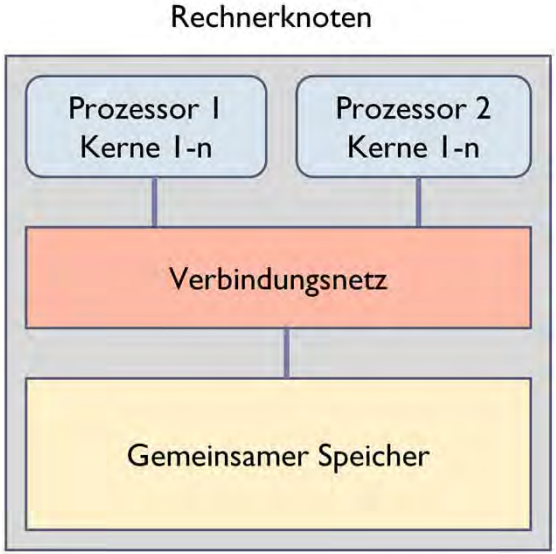
Mehrprozessorsysteme mit verteiltem Speicher



- Prozesse sehen nur den Adressraum im lokalen Speicher
- Leistungssteigerung:

  Dasselbe Programm läuft parallel auf allen Prozessoren; seine Daten sind auf die lokalen Speicher der Rechnerknoten aufgeteilt
- In dieser klassischen Form ausgestorben

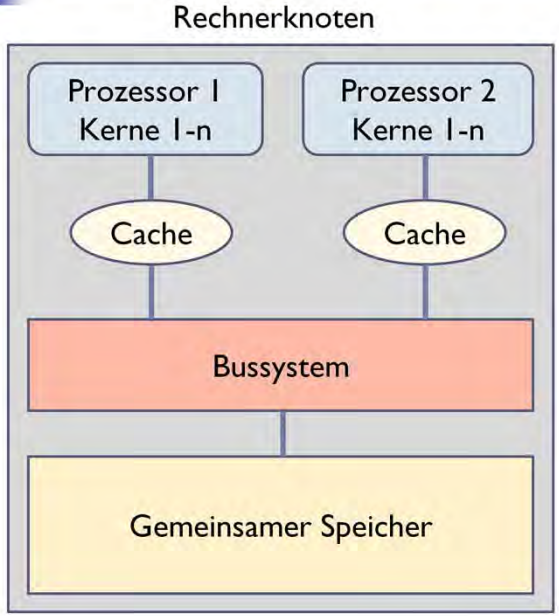
Mehrprozessorsysteme mit gemeinsamem Speicher



- Jeder Prozess/Thread sieht den gesamten Adressraum des gemeinsamen Speichers
- Leistungssteigerung:

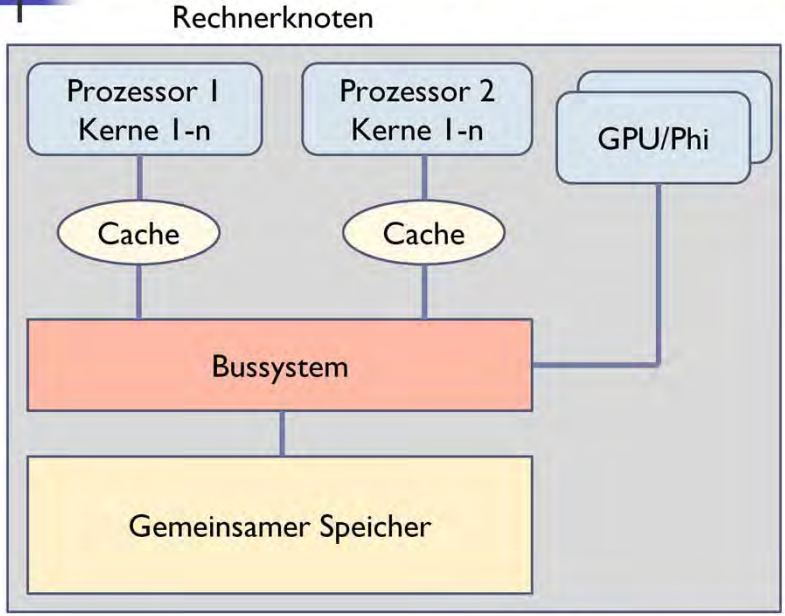
  Dasselbe Programm läuft parallel auf allen Prozessoren; seine Daten sind auf im gemeinsamen Speicher für alle zugreifbar

Gemeinsamer Speicher und Cache



- In der Realität immer auch mehrstufige Cache-Speicher
- Sehr komplex mit Konsistenz und Kohärenz
- Neue Fragen der Prozessorzuteilung treten auf (Scheduling)

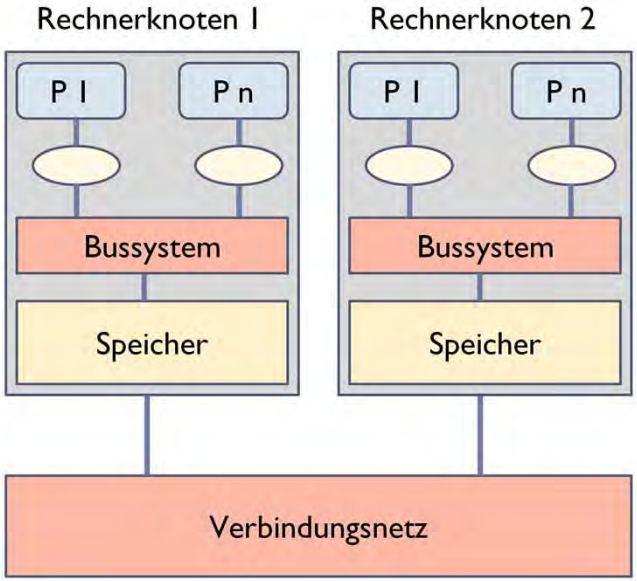
Gemeinsamer Speicher und Beschleuniger



Weitere Leistungssteigerung (und Stromeinsparung) durch Beschleunigerkarten

- General purpose computing on graphics processing units (GPGPU)
- Intel Xeon Phi

Mischform gemeinsamer/verteilter Speicher (Standard)



Existierende HLR sind heute meist eine Kombination aus Rechnerknoten mit gemeinsamem Speicher, von den man viele verwendet und sie über ein Verbindungsnetz verbindet

Vor- und Nachteile der Ansätze

- Mehrprozessorsysteme mit verteiltem Speicher
  - Hohe Ausbaubarkeit (>100.000 Prozessoren)
  - Komplexe Programmierung (Nachrichtenaustausch)
  - Reine Variante existiert aber nicht mehr, nur Mischformen mit gemeinsamem Speicher

- Mehrprozessorsysteme mit gemeinsamem Speicher
  - Geringe Ausbaubarkeit (einige dutzend Prozessorkerne und/oder Prozessoren)
  - „Einfachere" Programmierung (Verwendung gemeinsamer Speicherbereiche)

Weitere Bezeichnungen

Verteilter Speicher

- Multicomputersystem
- Schwache Kopplung
- Lose Kopplung
- Massiv paralleles System
- MPP – massive parallel processing

Gemeinsamer Speicher

- Multiprozessorsystem
- Multi-core system
- Enge Kopplung
- SMP – symmetric multiprocessing
- Mit Beschleunigern
  - Many-core system

Abgrenzungen

Verteilter Speicher

- Rechnerknoten pro HLR:
  - O(100)-O(10.000)
- Kommunikation:
  - Nachrichtenaustausch
- Betriebssysteme:
  - eine Instanz pro Knoten

Gemeinsamer Speicher

- Prozessorkerne pro Rechnerknoten:
  - O(10)
- Kommunikation:
  - gemeinsame Variable
- Betriebssystem:
  - eine Instanz

4. Skalierbarkeit

„Skalierbarkeit" nirgends eindeutig definiert, aber der wohl am häufigsten benutzte Begriff beim Hochleistungsrechnen

Gemeint ist: Ausbaubarkeit unter Beibehaltung gewisser positiver Charakteristika

- Z.B. Ein Programm skaliert gut, wenn es bei großer Prozess- oder Threadzahl noch hohe Leistung bringt
- Ein Netz skaliert gut, wenn beim Ausbau die Leistung mit dem investierten Geld korreliert

5. Verbindungsnetze und Topologien

- Im einfachsten Fall
  - Gemeinsamer Speicher: Bussystem
  - Verteilter Speicher: Sterntopologie mit Switch
- Im komplexen Fall
  - Alle Varianten, jedoch keine Vollvernetzung

Probleme

- Latenzzeiten, Übertragungszeiten
- Netzbelastung, Kollisionen

Beispiele von Verbindungsnetzen

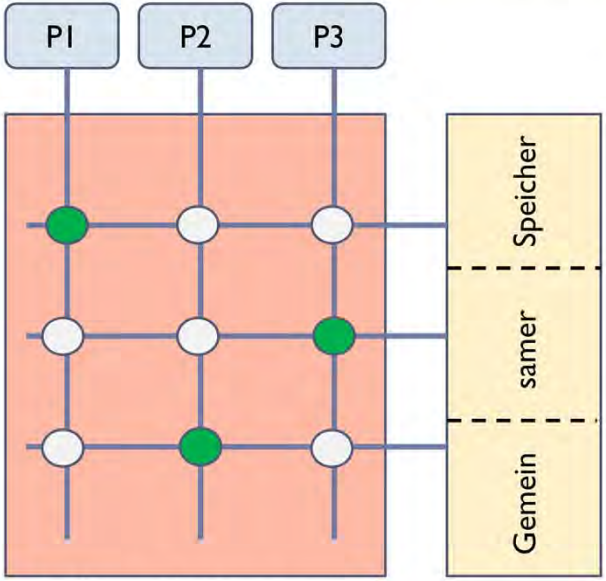
Es gibt hier eine Vielzahl von Konzepten !

- Immer wieder neue Konzepte mit der Werbung
  „das beste je entwickelte Netzwerk"

Wir greifen drei davon zur Illustration heraus:

- Kreuzschienenverteiler
- Zweidimensionaler Torus
- Hypercube

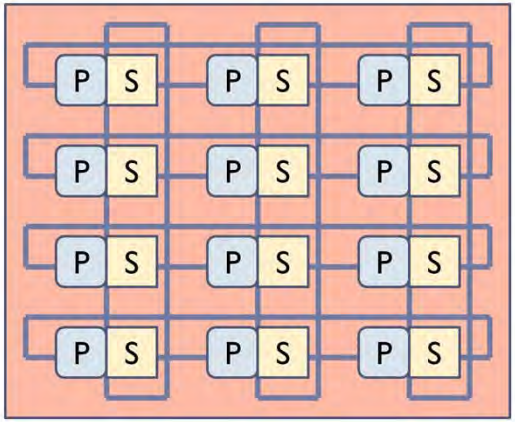
Verbindungsnetz bei gemeinsamem Speicher



3x3-Kreuzschienenverteiler

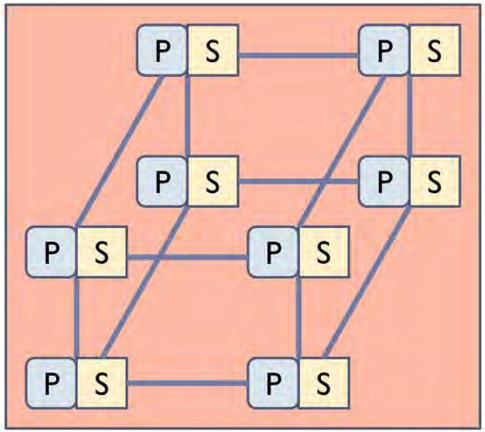
- Kreuzschienenverteiler n x m
- Im günstigsten Fall wie ein m-Bus-System
- Hoher technischer Aufwand
- Reduktion der Konflikte auf dem Bus
- Verwendet zwischen Prozessorkernen im Prozessor

Verbindungsnetz bei verteiltem Speicher (1)



- Zweidimensionaler Torus/Array
- Konstante Nachbarschaft, deshalb beliebig erweiterbar
- Entfernungsabhängige Übertragungszeiten
- Knotenzahl vervierfacht, maximaler Pfad verdoppelt sich

Verbindungsnetz bei verteiltem Speicher (2)



- Hypercube (n-dimensionaler Binärer Würfel)
- #Nachbarn = Dimension
  - Für technische Umsetzung problematisch
- Kurze maximale Entfernungen
- Hoher Grad der Vernetzung
- Knotenzahl vervierfacht, maximaler Pfad wächst um zwei

6. Hintergrundspeicher

- Lokale Platte an jedem Rechnerknoten
  - Heute meist nur für Servicezwecke auf dem Rechnerknoten
- Dateiserver ins Netz eingebunden
  - Persistente Datenhaltung
  - Engpass bei Datenzugriff
- Storage Area Network (SAN)
  - Speicherkomponenten mit eigenem Netz an die Komponenten des Clusters angehängt
- Hierarchical Storage Management (HSM) und Bandarchive

Ein-/Ausgabe war bisher vernachlässigte Fragestellung – jetzt intensiver untersucht

7. Spezialkonzepte

Historische Architekturen

- Workstationcluster
- Gridcomputing

Aktuelle Architekturen

- Cloudcomputing

Workstationcluster

- Cluster of workstations (COW)
- Network of workstations (NOW)

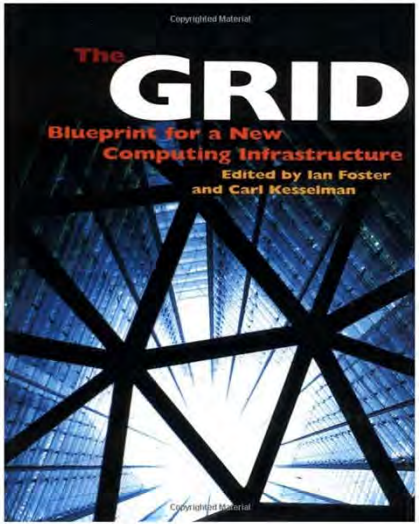
- Beowulf cluster (Sterling et al.)
  - Nur Standardkomponenten (commodity of the shelf components, COTS)

    Intel/AMD, Ethernet, Linux

Der Arme-Leute-Parallelrechner

Helics-Cluster Uni Heidelberg 2001

Gridcomputing



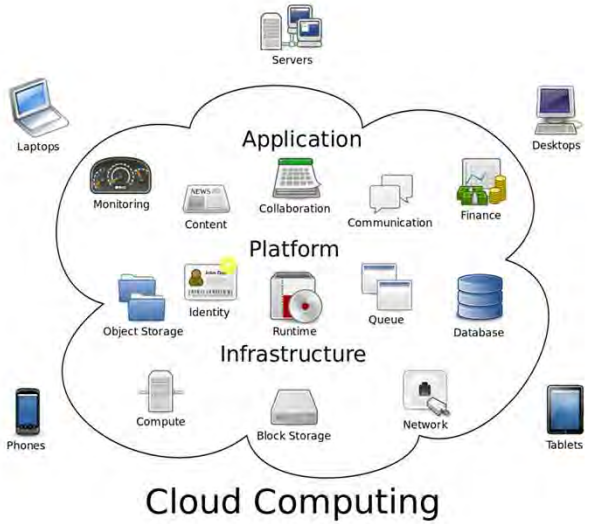
- Jederzeit verfügbare (hohe) Rechenleistung
  - Vergleichbar zu Elektrizität heute
- Netz von Hochleistungsrechnern

Der Superrechner der reichen Leute

Neues Konzept ab ca. 1999, aber:

- kam nie so richtig zum Fliegen trotz vieler Millionen von Forschungs-mitteln weltweit

Cloudcomputing



- Jederzeit verfügbare (hohe) Leistung zum Rechnen, Speichern, Programmenutzen …
- Netz von IT-Komponenten

Der Rechner/Speicher für die Zukunft ?

Hardware-Architekturen
Zusammenfassung

- Leistungssteigerung durch Parallelismus
- Erste wichtige Begriffsbildung durch Flynn
- Wir unterscheiden Architekturen mit verteiltem und mit gemeinsamem Speicher
- Die Skalierbarkeit ist bei verteiltem Speicher sehr hoch, dafür erschwert sich die Programmierbarkeit
- Reale Hochleistungsrechner sind meist viele vernetzte Rechnerknoten mit jeweils gemeinsamem Speicher und mehreren Mehrkernprozessoren
- Verbindungsnetze gibt es mit vielen Topologien
- Speichersysteme nutzen ebenfalls Parallelität

Hardware-Architekturen
Die wichtigsten Fragen

- Auf welchen Ebenen finden wir Parallelismus?
- Wie unterteilt Flynn die Rechnerarchitekturen?
- Wie funktionieren Systeme mit verteiltem Speicher?
- Wie funktionieren Systeme mit gemeinsamem Speicher?
- Welche Vor- und Nachteile haben die Ansätze?
- Wie sind reale Systeme aufgebaut?
- Welche Aufgabe hat das Verbindungsnetz und wie ist es strukturiert?
- Welche Konzepte finden wir beim Hintergrundspeicher?
- Welche weiteren Architekturen finden wir im Umfeld?