Hardware-Architekturen

- Parallelismus
- Klassifikation nach Flynn
- Erweiterung: die Sicht auf den Speicher
- Mehrprozessorsysteme mit verteiltem Speicher
- Mehrprozessorsysteme mit gemeinsamem Speicher
- Diskussion der beiden Ansätze
- Skalierbarkeit
- Verbindungsnetze und Topologien
- Betriebssystemaspekte

Hardware-Architekturen
Die zehn 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 Betriebssysteme finden wir bei HLR ?
- Welche weiteren Architekturen finden wir im Umfeld ?

Parallelismus – Die Sache mit den Ochsen

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

Seymour Cray (1925-1996)

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

W. Gropp, E. Lusk, A. Skjellum

- Wir erzielen höhere Leistung durch die parallele Nutzung leistungsschwächerer Einzelkomponenten
- Hochleistungsrechner sind immer Parallelrechner

Die beiden Zitate sind dem Buch von Bauke/Mertens über Cluster-Computing entnommen.

Ebenen des Parallelismus im Rechner

- Parallele Rechnerarchitekturen
  - Besitzen Verarbeitungseinheiten die koordiniert gleichzeitig an einer Aufgabe arbeiten
- Verarbeitungseinheiten
  - Spezialisierte Einheiten wie z.B. Pipelines
  - Gleichartige Rechenwerke
  - Prozessorkerne
  - Prozessoren
  - Vollständige Rechner
  - Hochleistungsrechnersysteme

Im Speichersystem finden wir die Existenz vieler Festplatten und vieler Bandlesegeräte.

In der Vernetzung finden wir parallele Wege zwischen vielen Paaren von Kommunikationspartnern.

Flynnsche Klassifikation

Klassifikation nach Flynn (1972)

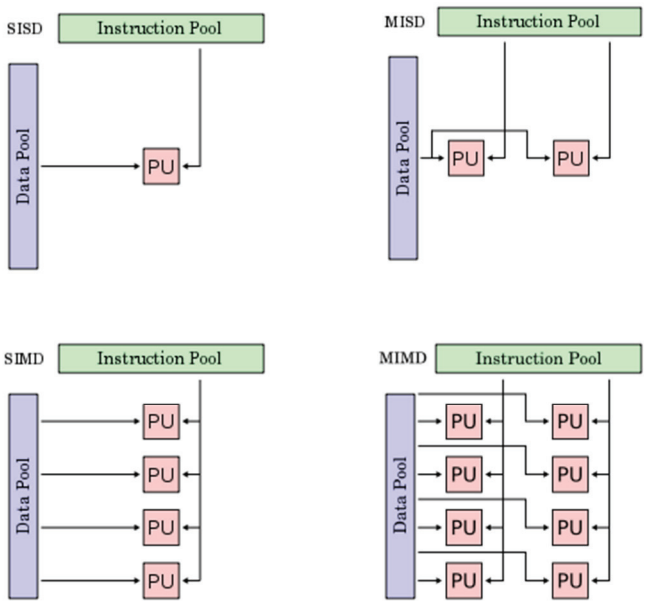
- Rechner arbeiten mit Befehlsströmen und Datenströmen
- Aus ihrer Kombination ergeben sich 4 Varianten

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

Siehe: http://en.wikipedia.org/wiki/Flynn%20s_Taxonomy

Die Klassifikation ist historisch. Sie dient hier einer ersten Unterteilung unserer Rechner in verschiedene Klassen, genügt aber nicht den aktuellen Anforderungen und kann die aktuellen Systeme nicht adäquat erfassen.

Flynnsche Klassifikation...



Quelle: Wikimedia Commons (http://en.wikipedia.org/wiki/File:SISD.svg, http://en.wikipedia.org/wiki/File:MISD.svg, http://en.wikipedia.org/wiki/File:SIMD.svg, http://en.wikipedia.org/wiki/File:MIMD.svg)

Flynnsche Klassifikation...

Was ist was bei Flynn?

- SISD: klassische von-Neumann-Architektur Monoprozessor-Rechner
- SIMD: Vektorrechner und Feldrechner
- MISD: diese Klasse ist leer
- MIMD: alles, was uns interessiert: die Mehrprozessorsysteme

Wir müssen die Klasse der MIMD-Rechner weiter aufgliedern

Unterteilung von Flynns MIMD-Klasse

Unterteilung der Flynnschen Systeme

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

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

Moderne Prozessoren enthalten jetzt fast immer mehrere Prozessorkerne, die wie Prozessoren geringerer Leistungsfähigkeit arbeiten. Dies verkompliziert unsere Betrachtungen. Später mehr dazu.

Siehe: http://en.wikipedia.org/wiki/Multi_core

Ein Prozeß ist ein auf einem Prozessor ablaufendes Programm. Ein Prozessor mit z.B. vier Prozessorkernen kann vier Prozesse echt gleichzeitig abarbeiten. Daneben werden ja auf jedem Einzelprozessor normalerweise auch mehrere Prozesse zeitlich verzahnt (also quasi-gleichzeitig) abgearbeitet.

Unterteilung von Flynns MIMD-Klasse...

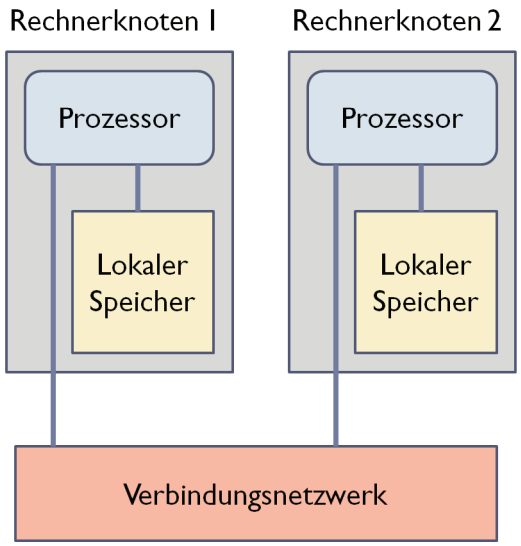
Unterscheidungskriterien

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

Klassen

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

Mehrprozessorsysteme mit verteiltem Speicher

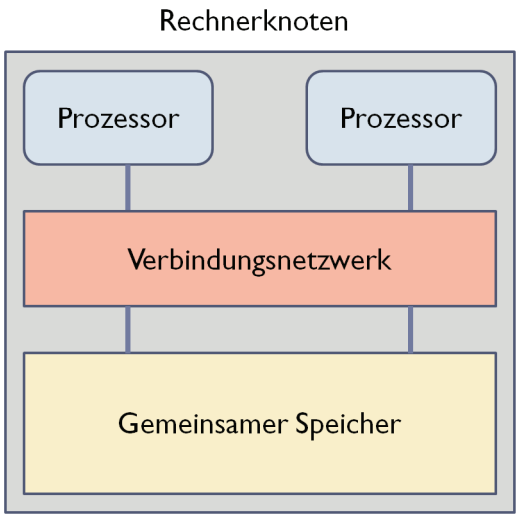


- Prozesse sehen nur den Adreßraum im lokalen Speicher
- Leistungssteigerung:

  Dasselbe Programm läuft parallel auf allen Prozessoren; seine Daten sind auf die lokalen Speicher der Rechnerknoten aufgeteilt

Im uns am besten bekannten Fall ist der Rechnerknoten ein einzelner Rechner (z.B. PC) und das Verbindungsnetzwerk ein Ethernet-Netzwerk. Bei Hochleistungsrechnern ist der Rechnerknoten ein Einschub in einem Rack (oder auch nur ein Teil eines solchen Einschubs) und das Verbindungsnetzwerk ist etwas hochspezielles.

Mehrprozessorsysteme mit gemeinsamem Speicher

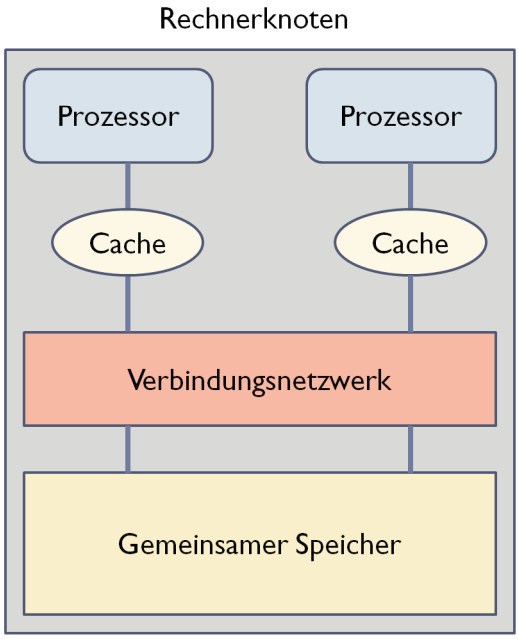


- Jeder Prozeß sieht den gesamten Adreßraum des gemeinsamen Speichers
- Leistungssteigerung:

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

Im uns am besten bekannten Fall handelt es sich hier um einen Rechner mit einem Prozessor und z.B. zwei Prozessorkernen, wie wir ihn heute als normalen PC kaufen. Das Verbindungsnetz ist hier der Prozessor-Speicher-Bus im Rechner. Bei Hochleistungsrechnern finden wir komplexe Formen der Spezialhardware für das Verbindungsnetz.

Mehrprozessorsysteme mit gemeinsamem Speicher und Cache (!)



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

Im uns am besten bekannten Fall handelt es sich hier um einen Rechner mit einem Prozessor und z.B. zwei Prozessorkernen, wie wir ihn heute als normalen PC kaufen. Das Verbindungsnetz ist hier der Prozessor-Speicher-Bus im Rechner. Bei Hochleistungsrechnern finden wir komplexe Formen der Spezialhardware für das Verbindungsnetz.

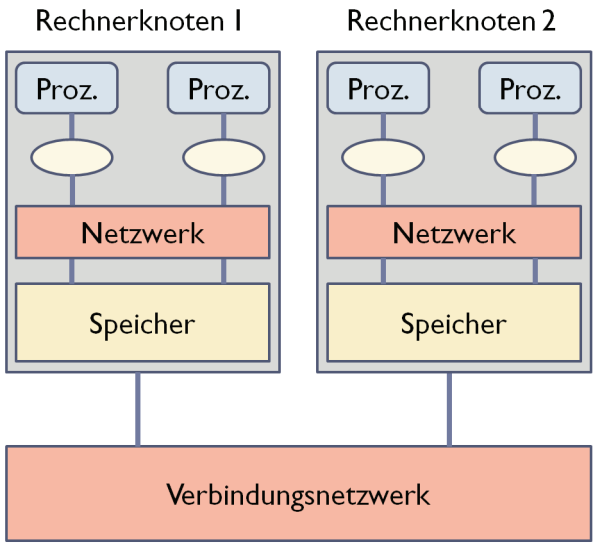
Vor- und Nachteile der Ansätze

- Mehrprozessorsysteme mit verteiltem Speicher
  - Hohe Ausbaubarkeit (100.000+ Prozessoren)
  - Komplexe Programmierung (Nachrichtenaustausch)
- Mehrprozessorsysteme mit gemeinsamem Speicher
  - Geringe Ausbaubarkeit (einige dutzend Prozessorkerne oder Prozessoren)
  - „Einfachere“ Programmierung (Verwendung gemeinsamer Speicherbereiche)

Die Programmierung von Systemen mit gemeinsamem Speicher ist nur auf den ersten Blick einfacher. Soll maximale Effizienz erzielt werden, so ist das auch beliebig schwierig. Allerdings gibt es in diesem Bereich halbwegs brauchbare automatische Ansätze durch Compiler-Unterstützung. Künftig wird man Kenntnisse der Programmierung dieser Architekturen vermehrt brauchen, wenn nämlich die Mehrkernprozessoren sich weiter verbreiten.

Ausbauen kann man die Systeme natürlich beliebig – die Frage ist, wie lange die Leistungssteigerung den aufgewendeten Finanzen folgt.

Reale Systeme: Alles in Einem



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

Weitere Bezeichnungen

Verteilter Speicher

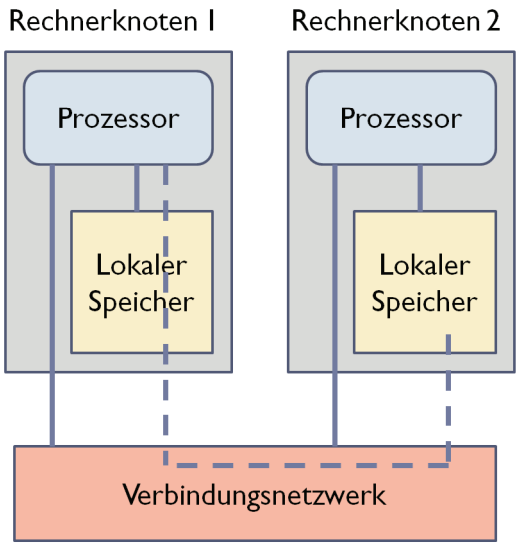
- Multicomputersystem
- Schwache Kopplung
- Lose Kopplung
- Massiv paralleles System
- MPP – massive parallel processing

Gemeinsamer Speicher

- Multiprozessorsystem
- Enge Kopplung
- SMP – symmetric multiprocessing

SMP ist eine Bezeichnung aus der Betriebssystemtechnik. Wir kommen später darauf zurück.

Mischform: Verteilter gemeinsamer Speicher



- Logisch sieht jeder Prozeß den gesamten Adreßraum aller aggregierten lokalen Speicher
- Physikalisch ist der Adreßraum verteilt
- Bereitstellung durch Hardware und/oder Software
- Bezeichnung auch: DSM (distributed shared memory)

DSM-Architekturen kommen im alltäglichen PC-Bereich nicht vor und spielen auch bei Hochleistungsrechnern keine Rolle mehr.

Modellierung bzgl. Zugriffszeiten

- UMA: uniform memory access model
  - Gemeinsamer Speicher
- (cc)NUMA: (cache coherent) non uniform memory access model
  - Verteilter gemeinsamer Speicher (mit Cache-Kohärenz)
- NORMA: no remote memory access model
  - Verteilter Speicher
- (N)UCA: (non) uniform communication architecture model
  - Nicht uniform: z.B. Cluster von SMP-Maschinen

Am gebräuchlichsten in Prospekten ist die Bezeichnung (cc)NUMA. Die anderen Begriffe kommen seltener vor und dann zur Abgrenzung gegenüber NUMA.

Der Begriff 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 Prozeßzahl noch hohe Leistung bringt
- Ein Netzwerk skaliert gut, wenn beim Ausbau die Leistung mit dem investierten Geld korreliert

Siehe: http://en.wikipedia.org/wiki/Scalability

Verbindungsnetze

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

Probleme

- Latenzzeiten, Übertragungszeiten
- Netzbelastung, Kollisionen

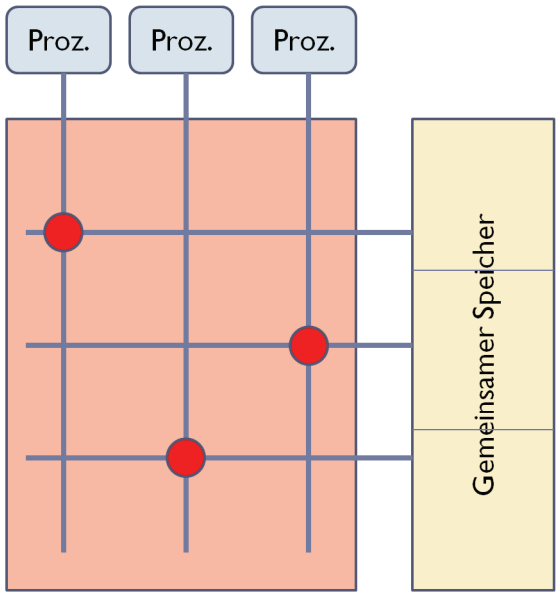
Beispiele von Verbindungsnetzen

Es gibt hier eine Vielzahl von Konzepten !

Wir greifen drei davon zur Illustration heraus:

- Kreuzschienenverteiler
- Zweidimensionaler Torus
- Hypercube

Verbindungsnetz bei gemeinsamem Speicher



3x3-Kreuzschienenverteiler

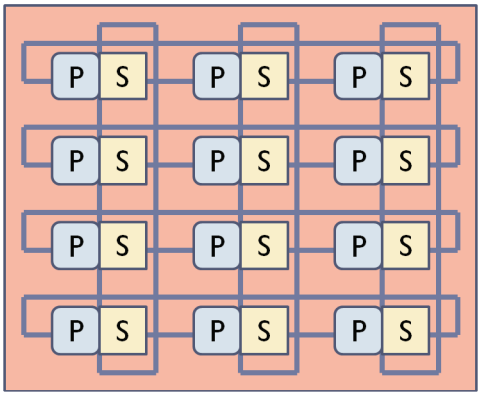
- Kreuzschienenverteiler n x m
- Im günstigsten Fall wie ein m-Bus-System
- Hoher technischer Aufwand
- Reduktion der Konflikte auf dem Bus

Engl.: crossbar switch

Siehe: http://en.wikipedia.org/wiki/Crossbar_switch

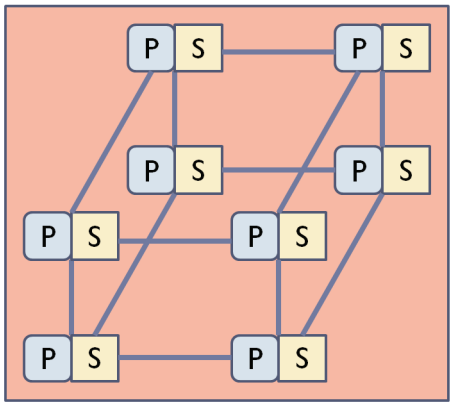
Verbindet Prozessoren mit Speichermodulen; die Module (hier drei) können unabhängig voneinander angesprochen werden.

Verbindungsnetz bei verteiltem Speicher (1)



- Zweidimensionaler Torus/Array
- Konstante Nachbarschaft, deshalb beliebig erweiterbar
- Entfernungsabhängige Übertragungszeiten
- Knotenzahl verdoppelt, maximaler Pfad wächst stark an

Verbindungsnetz bei verteiltem Speicher (2)



- Hypercube (n-dimen. Binärer Würfel)
- #Nachbarn = Dimension
- Kurze maximale Entfernungen
- Hoher Grad der Vernetzung
- Knotenzahl doppelt, maximaler Pfad wächst um eins

Hintergrundspeicher

- Lokale Platte an jedem Rechnerknoten
  - Heute meist nur zur Zwischenspeicherung
- Dateiserver ins Netz eingebunden
  - Persistente Datenhaltung
  - Flaschenhals bei Datenzugriff
- Storage Area Network (SAN)
  - Speicherkomponenten mit eigenem Netz an die Komponenten des Clusters angehängt
- Bandarchive

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

Betriebssysteme

- Betriebssysteme für Hochleistungsrechner
  - Auf den Rechnerknoten:
    Fast immer Unix-Derivate (meist Linux), selten Windows
  - Über alle Knoten hinweg
    Zusatzsoftware, die den Verbund nutzbar macht
    (nicht in das Betriebssystem integriert)
- Betriebssystemforschung
  - Single System Image (SSI)
    - Das System erscheint dem Anwender wie ein System mit einem Knoten
    - Lokalisierung von Diensten verborgen

SSI hat in der Praxis keine Relevanz. Das Verbergen eines konkreten Ausführungsortes taucht aber als Forschungsziel immer wieder auf, zuletzt beim Cloud-Computing.

Spezialkonzept: Rechnercluster

- Cluster of workstations (COW)
- Network of workstations (NOW)
- Beowulf cluster (Sterling et al.)
  - Nur Standardkomponenten (commodity of the shelf components, COTS)

    Pentium, Ethernet, Linux

Der Arme-Leute-Parallelrechner

Spezialkonzept: Grid

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

Der Superrechner des reichen Mannes ☹

Konzept kam nie so richtig zum Fliegen trotz vieler Millionen von Forschungsmitteln weltweit

Spezialkonzept: Cloud

- Jederzeit verfügbare (hohe) Leistung zum Rechnen, Speichern, Programmenutzen ...
- Netzwerk von IT-Komponenten

Der Rechner für die Zukunft ?

2020: Konzept kam nie so richtig zum Fliegen trotz vieler Millionen von Forschungsmitteln weltweit ?

Abgrenzungen

	Verteilter Speicher	Gemeinsamer Speicher	Cluster	Verteiltes System
#Prozessoren	O(100) – O(100.000)	O(10)	O(10) – O(10.000)	O(10) – O(1000)
Kommuni-kation zwischen Prozessen	Nachrichten	Gemeinsame Variable	Nachrichten	RPC, Nachrichten, Middleware
Single System Image	Selten	Immer	Selten	Nie
Betriebs-system	Sparversion	SMP-BS	Unix homogen	Verschiedene heterogen
Besitzer	Einer	Einer	Einer oder mehrere	Mehrere

Hardware-Architekturen
Zusammenfassung

- 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 Mehrkernprozessoren
- Verbindungsnetze gibt es mit vielen Topologien
- Speichersysteme nutzen ebenfalls Parallelität
- Als Betriebssystem kommt meist Linux zum Einsatz