Kapitel 5

Parallelverarbeitung

- Formen der Parallelität
- Klassifikation von parallelen Rechnerarchitekturen
- Exkurs über Verbindungsstrukturen

Formen der Parallelität

- Parallelität auf Bitebene: bis etwa 1985
  - **Kombinatorische Addierer** und **Multiplizierer**, etc.
  - → wachsende Wortbreite auf 64 Bit
- Parallelität auf Instruktionsebene: 1985 bis heute
  - **Pipelining** der Instruktionsverarbeitung
  - **Mehrere Funktionseinheiten** (→ superskalare Prozessoren) bei mehr als 4 Funktionseinheiten werden Datenabhängigkeiten oft zum Hindernis für eine effiziente Ausnutzung
  - **Vektorprozessoren** führen eine Operation parallel auf vielen Daten durch (Bsp: Cray 1 [1974])
- Parallelität auf Prozessor-/Rechnerebene
  - nur so scheinen Beschleunigungen um Faktor 50 und mehr möglich

Ein Beispiel

Computer Zeitung Nr. 45 / 9.11.2000

NUMBERCRUNCHERS

IBMs Asci White entreißt Intel die Supercomputing-Krone

Dallas (fin) – Mit dem Massiv-Parallel-System Asci White hat IBM Intels Asci Red als weltschnellsten Computer verdrängt. Auch sonst dominiert „Big Blue" die Numbercruncher-Szene wie nie zuvor.

Nach über drei Jahren ist der Dauerspitzenreiter der Top-500-Rangliste der Supercomputer, Intels Asci Red, entthront. Neue Nummer 1 des von den Wissenschaftlern Hans Werner Meuer, Erich Strohmaier und Jack Dongarra halbjährlich erstellten Rankings ist IBMs Asci White, ein an der US-Forschungseinrichtung Lawrence Livermore Lab installierter 160-Tonnen-Koloß mit 8192 Power-3-Prozessoren und der Rekordrechenleistung von 4,9 Teraflops (Billionen Gleitkommaoperationen pro Sekunde) im Linpack-Benchmark.

Überhaupt dominiert Big Blue mit den RS/6000-SP-Systemen die Top-500-Liste nach Belieben: Mit 215 Systemen stammen 43 Prozent aller dort vertretenen Rechner aus Armonk; vor 6 Monaten waren es lediglich 29 Prozent gewesen. Von den zehn schnellsten Rechnern der Welt tragen fünf das IBM-Logo.

Weiter auf dem Vormarsch sind die Cluster-Systeme: 112 Clustern aus SMP-Knoten (Symmetrisches Multiprocessing) mit vier oder mehr Prozessoren sind in der aktuellen Liste vertreten, da- zu 28 Systeme aus vernetzten Standard-Workstations. Dagegen haben reine SMP-Systeme keine Chance mehr: Ihr Anteil ist von 121 vor Halbjahresfrist auf ganze 17 geschrumpft.

Superrechner Marke Eigenbau

Zunehmender Beliebtheit erfreuen sich kostengünstige Selfmade-Rechner wie der Linux-Cluster Clic an der TU Chemnitz, der es mit seinen 528 Pentium-III-Prozessoren immerhin auf Platz 126 der Top 500 geschafft hat. Der Mannheimer Supercomputerexperte Hans Werner Meuer hält den Chemnitzer Eigenbau für den Rechner „mit dem besten Preis-Leistungs-Verhältnis weltweit". Insgesamt fünf solcher Selfmade-Systeme sind im aktuellen Ranking platziert, angeführt vom Alpha-Cluster Cplant der Sandia Labs auf Platz 86.

Die gebaute Rechenpower aller Top-500-Rechner ist im Vergleich zum Juni um 62 Prozent gestiegen und liegt jetzt bei 88,1 Teraflops. Damit wächst die Supercomputerleistung fast doppelt so schnell wie vom Moorschen Gesetz gefordert.

Größter Supercomputeranwender sind zwar immer noch die USA mit 241 Systemen, Europa hat jedoch deutlich aufgeholt und stellt jetzt 177 Numbercruncher.

Die Fläche von zwei Basketballfeldern Asci White für die Simulation von Na

Formen der **Parallelverarbeitung** auf Prozessor-/Rechnerebene

- **Arrayprozessoren**
  - Viele gleichartige Prozessoren, ev. mit lokalem Speicher
  - eine Steuereinheit für alle Prozessoren
  - gleiche Instruktionsfolge auf allen Prozessoren
  - **Beispiele:** Illiac IV (1972, 64 Knoten),
    CM-2 (1987, 65536 Knoten)

Nächster Schritt:

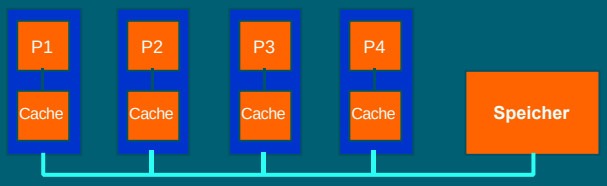
Mache Prozessoren in der Verarbeitung unabhängiger

Formen der **Parallelverarbeitung** auf Prozessor-/Rechnerebene

- **Multiprozessoren**
  - bestehen aus mehreren Prozessoren und haben einen gemeinsamen Speicher (shared memory).
  - Kommunikation erfolgt über den gemeinsamen Speicher
  - **Problem**: Evtl. Performanz-Probleme, wenn viele Prozessoren gleichzeitig auf den Speicher zugreifen wollen.
- **Multicomputer**
  - bestehen aus mehreren Rechnern ohne gemeinsamen Speicher.
  - jeder Prozessor mit eigenem Speicher (local memory)
  - Prozessoren tauschen Daten durch Nachrichten aus
  - **Problem**: Evtl. hoher Aufwand für Nachrichtenaustausch.

Beispiele für parallele Rechnerarchitekturen

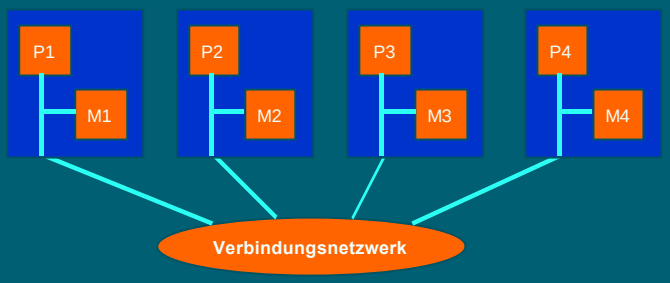
- Symmetrische Multiprozessoren (SMP) mit gemeinsamem nichtverteilten Speicher



- oft busbasiert mit physikalisch gemeinsamem Speicher
- homogene Prozessoren
- Reduktion der Speicherzugriffslatenz durch lokale Caches
- **Problem**: Cache-Kohärenz

Beispiele für parallele Rechnerarchitekturen

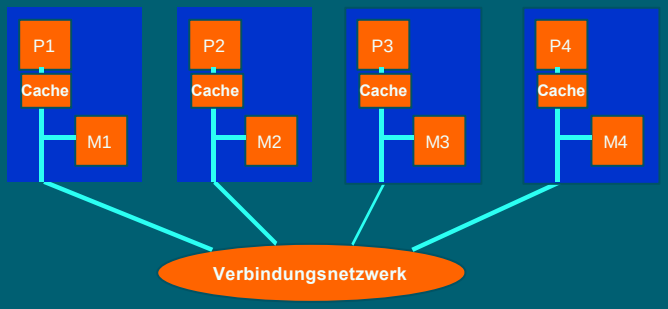
- Systeme mit Verbindungsnetzwerk und verteiltem gemeinsamen Speicher



- gemeinsamer Adreßraum, lokale Speichermodule
- **Beispiel:** Carnegie Mellon Cm* [1977]

Beispiele für parallele Rechnerarchitekturen

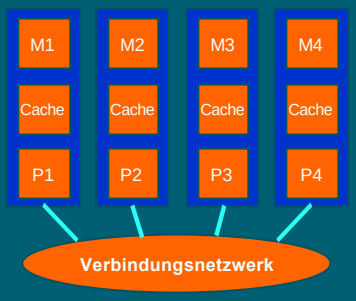
- Systeme mit Verbindungsnetzwerk, verteiltem gemeinsamen Speicher und lokalem Cache



- Verbindungsnetzwerk, z.B:. Crossbar Switch
- **Beispiel:** Sun Enterprise 10 000 [1998], SGI Origin 2000 [1993]
- Realisierungen mit bis zu 256 CPUs

Beispiele für parallele Rechnerarchitekturen

- Systeme mit Verbindungsnetzwerk und lokalem Speicher



- Zugriff auf fremden lokalen Speicher nicht möglich
- Kommunikation durch Austausch von Nachrichten
- gute physikalische Skalierbarkeit, mehrere Tausend CPUs möglich
- **Beispiele**: MPP (= Massively Parallel Processors), Cray T3E, IBM SP/2
  COW (= cluster of workstations)

Klassifikationsmerkmale bei parallelen Rechnerarchitekturen

Speichermodell:

- Nichtverteilter Speicher, gemeinsamer Adressraum
- verteilter Speicher, gemeinsamer Adressraum
- verteilter Speicher, getrennte Adressräume

Homogenität der Prozessoren

- **homogene Parallelrechner**: alle Prozessoren sind gleich
- **heterogene Parallelrechner**: Prozessoren dürfen sich hardwaremäßig unterscheiden

Hierarchie zwischen den Prozessoren

- **symmetrische Parallelrechner**: die Prozessoren sind bzgl. ihrer Rolle im System untereinander austauschbar
- **nichtsymmetrische Parallelrechner**: es gibt Masters und Slaves

Eigenständigkeit der Prozessoren

- **lose gekoppelte Parallelrechner**: Netzwerk von eigenständigen Rechnern
- **eng gekoppelte Parallelrechner**: physikalisch ein Rechner

Klassifikation nach Flynn [1966]

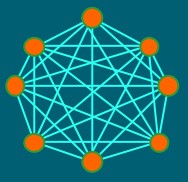
- **SISD** (**S**ingle **I**nstruction stream, **S**ingle **D**ata stream)
- **SIMD** (**S**ingle **I**nstruction stream, **M**ultiple **D**ata streams)
  - Die gleiche Instruktion wird parallel auf einen Vektor von Daten angewendet.
- **MISD** (**M**ultiple **I**nstruction streams, **S**ingle **D**ata stream)
  - mehrere Anweisungen parallel auf einem Datenstrom
- **MIMD** (**M**ultiple **I**nstruction streams, **M**ultiple **D**ata streams)

Kurzer Exkurs über Verbindungsstrukturen

- Der **Kommunikationsaufwand** zwischen den Prozessoren ist mitentscheidend für die Leistung von parallelen Rechnersystemen

  ⇒ nicht jedes Problem ist für Parallelisierung geeignet.

- Verschiedene Kommunikationsstrukturen unterscheiden sich hinsichtlich ihrer Kosten und ihrer Leistung.

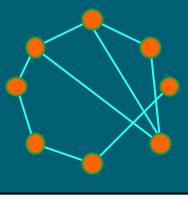


Beispiel:
Vollständiger Verbindungsgraph

Modellierung von Verbindungen

- Modellierung:
  Die Topologie eines Parallelrechners wird durch einen abstrakten Graphen G=(V,E) dargestellt mit
  - $V = \{ 1, \ldots , n \}$ — Menge der Knoten,
    sie repräsentieren die Prozessoren oder Switches
  - $E \subseteq \{ \{a,b\}; a,b \in V \}$ — Menge der Kanten,
    sie repräsentieren die Verbindungen

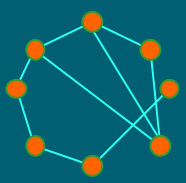
Beispiel:



Modellierung von Verbindungen ff

- Charakteristika einer Verbindungsstruktur
  - Durchmesser(G) = $\max_{v,w \in V}$ { Länge des kürzesten Pfades von v nach w}
  - Grad(G) = $\max_{v \in V}$ | { $w \in V$; $\{v,w\} \in E$ } |
  - Anzahl der Verbindungen (G) = | E |

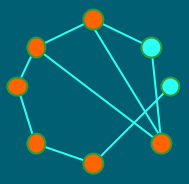
Beispiel:



Modellierung von Verbindungen ff

- Charakteristika einer Verbindungsstruktur
  - Durchmesser(G) = max $_{v,w \in V}$ { Länge des kürzesten Pfades von v nach w}
  - Grad(G) = max $_{v \in V}$ | { w ∈ V; {v,w} ∈ E } |
  - Anzahl der Verbindungen (G) = | E |

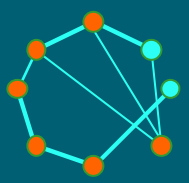
Beispiel:



Modellierung von Verbindungen ff

- Charakteristika einer Verbindungsstruktur
  - Durchmesser(G) = max $_{v,w \in V}$ { Länge des kürzesten Pfades von v nach w}
  - Grad(G) = max $_{v \in V}$ | { w ∈ V; {v,w} ∈ E } |
  - Anzahl der Verbindungen (G) = | E |

Beispiel:

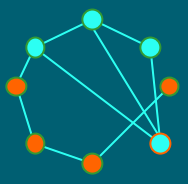


Durchmesser(G)=6

Modellierung von Verbindungen ff

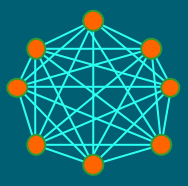
- Charakteristika einer Verbindungsstruktur
  - Durchmesser(G) = max $_{v,w \in V}$ { Länge des kürzesten Pfades von v nach w}
  - Grad(G) = max $_{v \in V}$ | { w ∈ V; {v,w} ∈ E } |
  - Anzahl der Verbindungen (G) = | E |

Beispiel:



Grad(G)=3

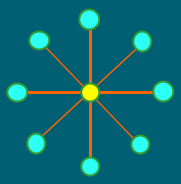
Vollständiger Verbindungsgraph

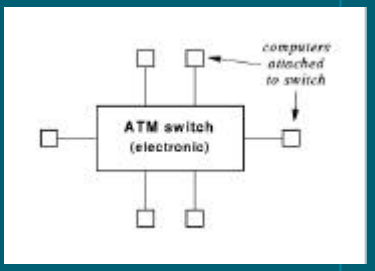


Zu teuer wegen dem großen Fanout

- Durchmesser = 1
- Grad = n-1
- |Verbindungen| = n(n-1)/2, keine „Kollisionen“

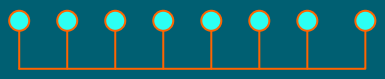
Stern





- Durchmesser = 2
- Grad = n-1
- |Verbindungen| = n-1

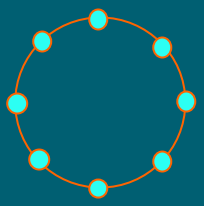
Bus



Topologie versagt bei den heutigen Technologien bei grossem Datentransfer zwischen den Prozessoren

- Ein Bus läßt sich als Stern modellieren, wobei der zentrale Knoten aber kein Prozessor ist, sondern den Bus repräsentiert
- Gleiches „Flaschenhalsproblem“ wie bei Stern

Ring



Beispiel für Ring: **Token-Ring**

Es kreist ein sogenanntes **Token** (spezielles Paket).
Ein Rechner darf nur dann senden, wenn er das Token besitzt.

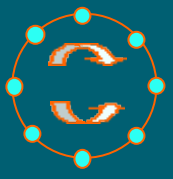
- Durchmesser = n/2
- Grad = 2
- |Verbindungen| = n

CDDI/FDDI-Ring

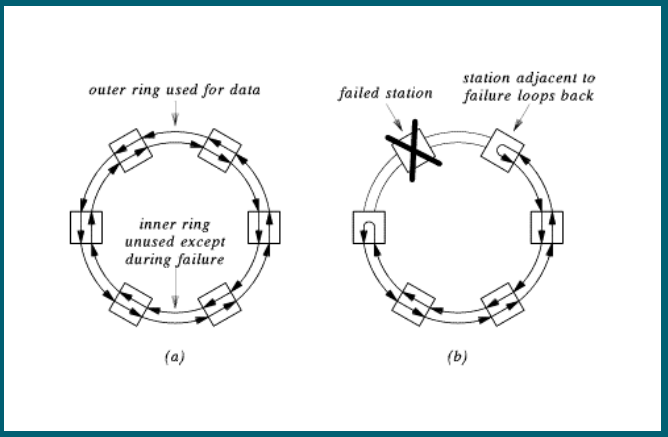
[Copper / Fiber Distributed Data Interconnect]

Charakteristika

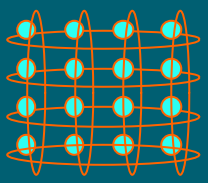
- Ring-Topologie
- Besteht aus zwei gegenläufigen Ringen
  → **Fehlertolerantes Netz**
- ... ansonsten wie beim Token-Ring



CDDI/FDDI-Ring ist fehlertolerant !



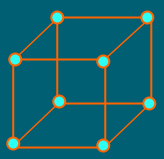
MESH („torusähnliches Gitter")



Wird z.B. bei MPPs benutzt

- Durchmesser = $n^{1/2}$
- Grad = 4
- |Verbindungen| = 2n

Hypercube (d-dimensionaler Würfel)



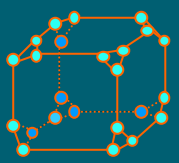
Bei vielen Prozessorne nicht verwendbar, da Grad zu hoch

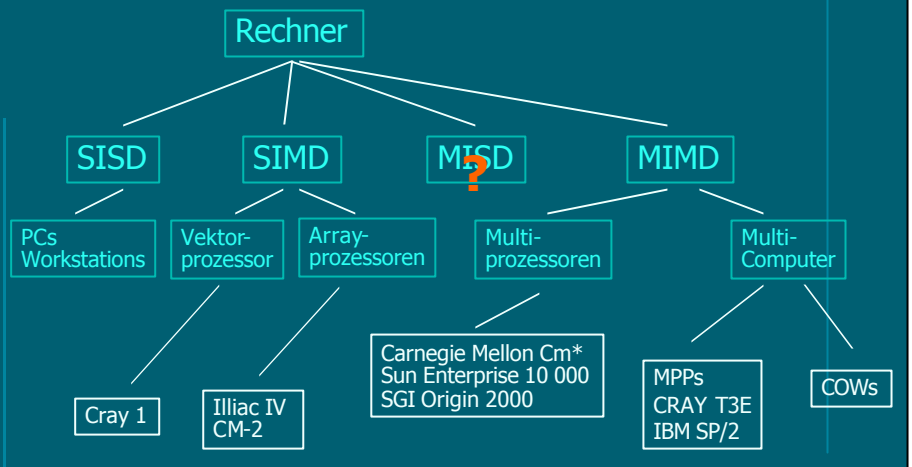
- Im Bild: d=3, n=8 (Knotenzahl)

Allgemein:

- Durchmesser = $\log_2 n$
- Grad = $\log_2 n$
- |Verbindungen| = $(n \log_2 n) / 2$, in der Regel keine Wartezeiten

Cube Connected Cycle (CCC)



- Prozessoren = $n \log_2 n$
- Durchmesser = $\lfloor 5/2 \log_2 n \rfloor - 1$
- Grad = 3
- |Verbindungen| = $3/2\ n \log_2 n$, in der Regel keine Wartezeiten

Zusammenfassung