Parallelrechner (13)

Metriken zur Performanzsteigerung

Hardwaremetriken

- *Verzögerung (Latenz)*
  - Für circuit switching: $T_S + p/b$ wobei T_S := Verbindungsaufbauzeit
    und p/b die Übertragungsverzögerung repräsentiert
    (p:=Paketgröße in bits, b:= (bilaterale) Bandbreite in bits/s)

  - Für packet switching: $T_a + n (p/b + T_d) + p/b$ wobei
    T_a := Zeit für die Zusammensetzung des Pakets,
    T_d := Verzögerung innerhalb eines Schalters
    n := Anzahl der Schalter

- *Bandbreite*
  - Gesamtbandbreite := Summe aller bilateralen Bandbreiten im Netz
  - Durchnittsbandbreite := durchschnittliche Outputkapazität der Netzwerkcontroller (Schnittstellen) der CPU´s

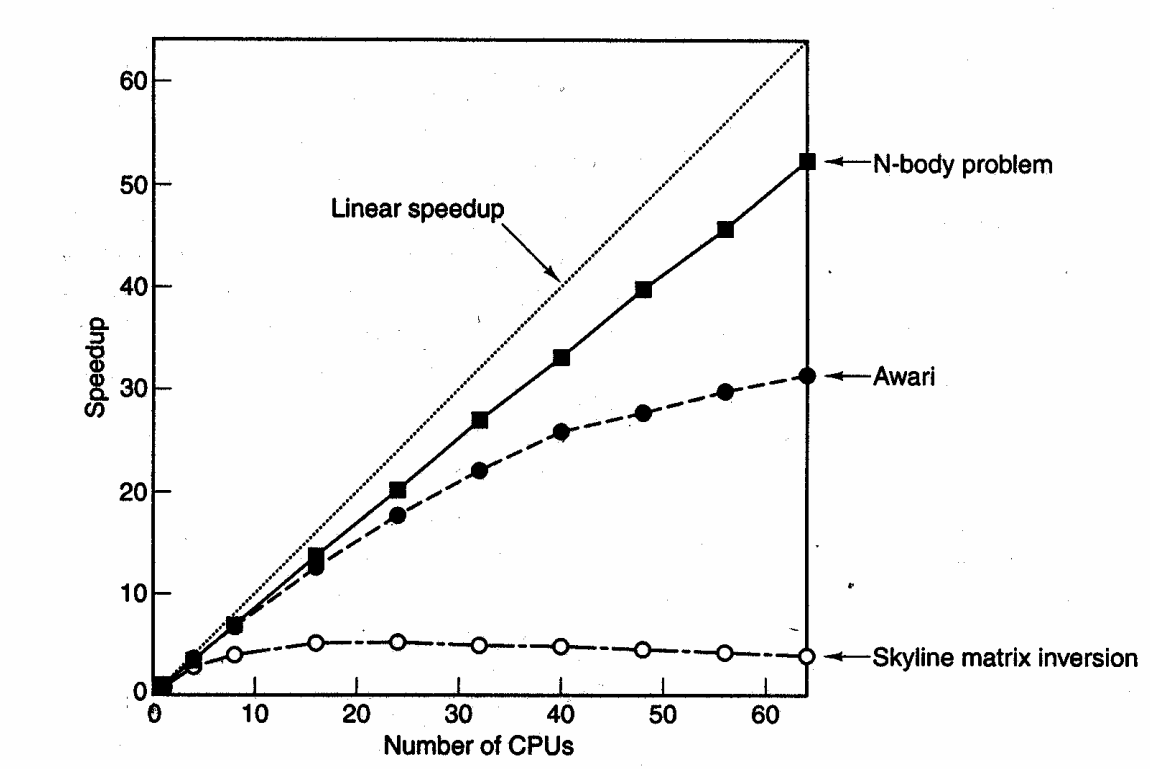
  Problem:
  Es gibt einen inhärenten trade-off zwischen den aus den Metriken ableitbaren Zielen
  - geringe Verzögerung bzw.
  - hohe Ausnutzung der Bandbreite

Parallelrechner (14)

Softwaremetriken

Schlüsselgröße (aus der Anwendersicht): *Geschwindigkeitssteigerung (speedup)*

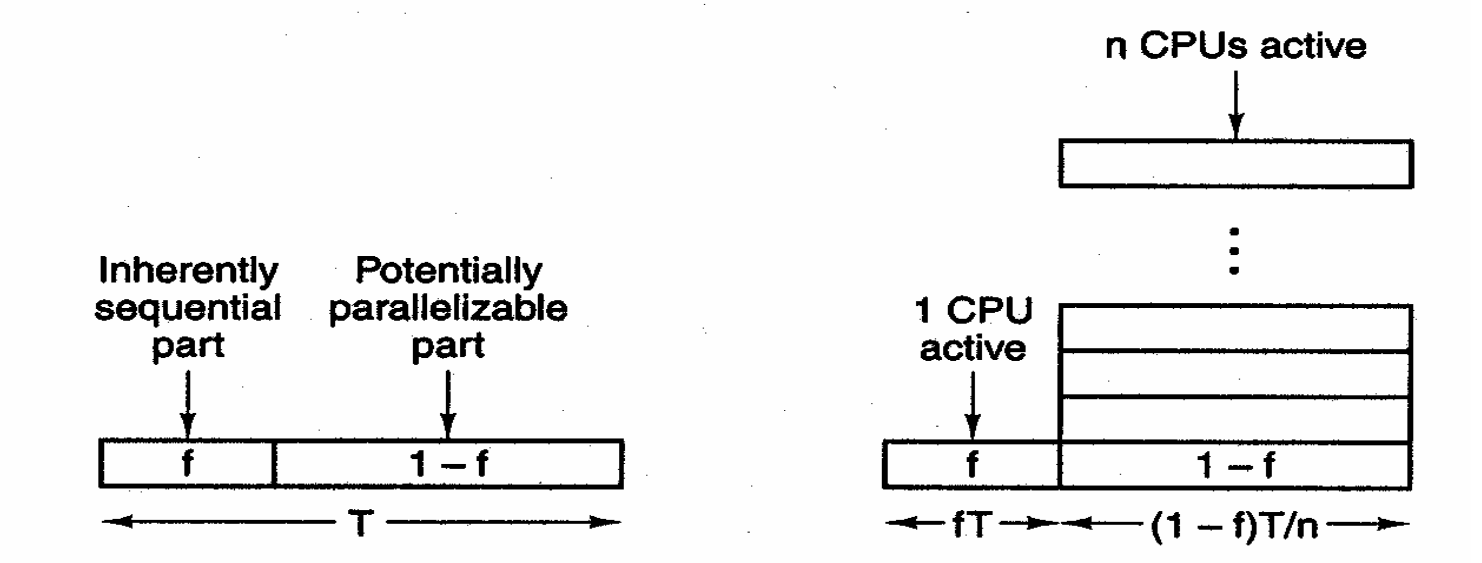
Beispielhafte graphische Darstellung:



Perfekter (optimaler) speedup := linearer speedup

Parallelrechner (15)

Herleitung des speedup:



Definition des speedup:

Speedup:= n/ (1 + (n-1) f) wobei

n:= Anzahl der CPU´s

f:= relative Anteil der Laufzeit des sequentiellen Codeanteils an der Gesamtlaufzeit

Amdahl´s Gesetz:

Perfekter speedup ist in der Realität nicht möglich, da f > 0 in der Regel.

Parallelrechner (16)

Methoden zur Performanzsteigerung

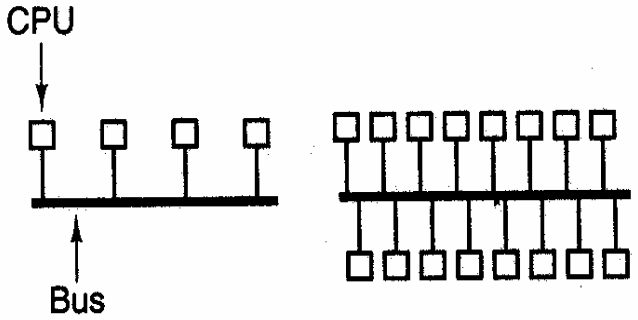
Einfachste Methode: Hinzufügen weiterer CPU´s

weitere Voraussetzung: Das Verbindungsnetzwerk ist entsprechend *skalierbar*

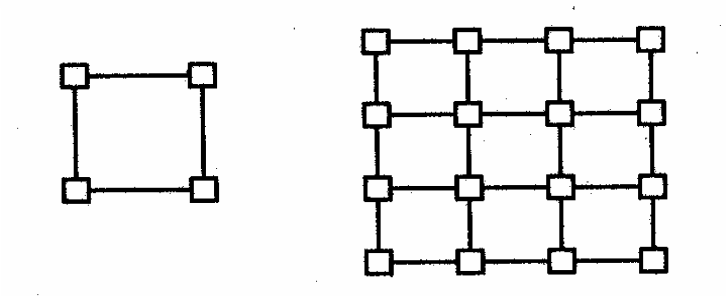
Skalierbarkeit ist gegeben, wenn effektiv mehr Rechnerleistung zur Verfügung steht, d.h. die mittlere Bandbreite pro CPU nicht fällt.

Beispiele:

nicht skalierbar:



skalierbar:



Parallelrechner (17)

Allerdings ist letzteres Beispiel **nicht skalierbar** bzgl. Verzögerung! (Durchmesser wächst!)
Wünschenswert wäre, dass die Verzögerung bei der Hinzufügung von CPU´s konstant bliebe.

Problem: Realität ist, Verzögerung wächst mit Größe (mindestens logarithmisch, wie beim hypercube)

Techniken zur Vermeidung größerer Datentransporte und damit längeren Programmausführungszeiten

- Data replication (z.B. caching, replicas)
- Prefetching (ausnutzen von räumlicher Lokalität)
- Multithreading

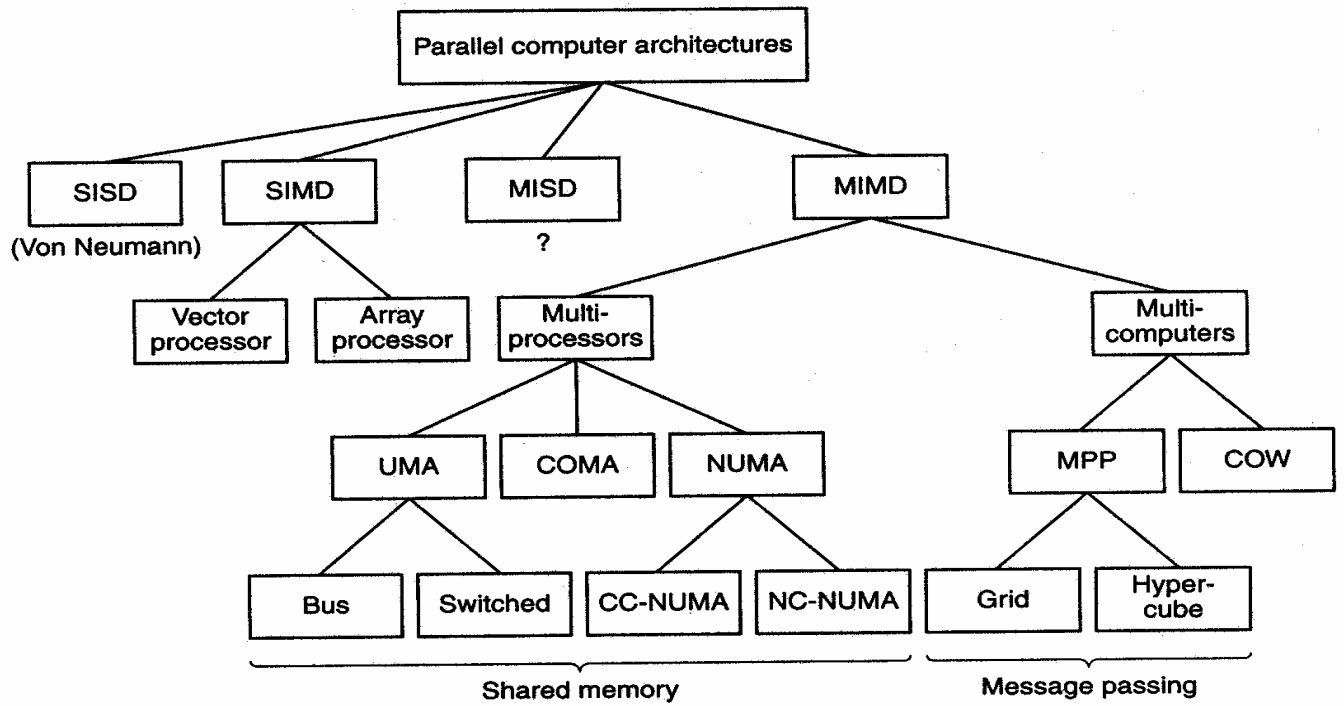
Taxonomie von Parallelrechnern

Flynn´s Klassifikationsschema:

Instruction streams	Data streams	Name	Examples
1	1	SISD	Classical Von Neumann machine
1	Multiple	SIMD	Vector supercomputer, array processor
Multiple	1	MISD	Arguably none
Multiple	Multiple	MIMD	Multiprocessor, multicomputer

Parallelrechner (18)

Erweitertes Klassifikationsschema:



UMA:= **Uniform Memory Access**

NUMA:= **NonUniform Memory Access**

COMA:= **Cache Only Memory Access**

MPP:= **Massively Parallel Processors**

COW:= **Cluster of Workstations**