

Universität Karlsruhe (TH)

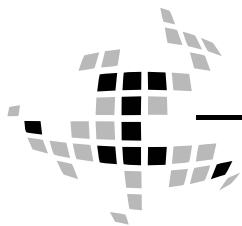
Forschungsuniversität · gegründet 1825

## Multikern-Rechner und Rechnerbündel

Prof. Dr. Walter F. Tichy

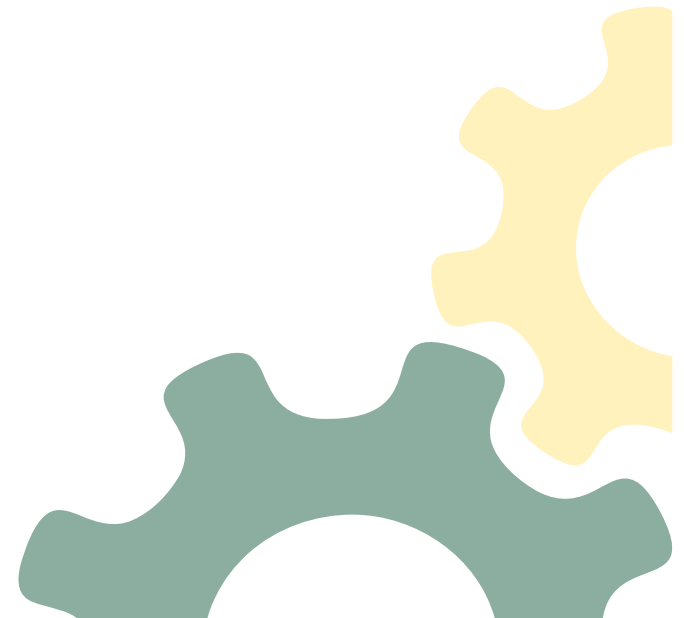
Dr. Victor Pankratius

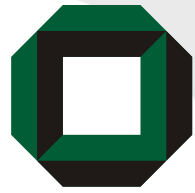
Ali Jannesari



Fakultät für **Informatik**

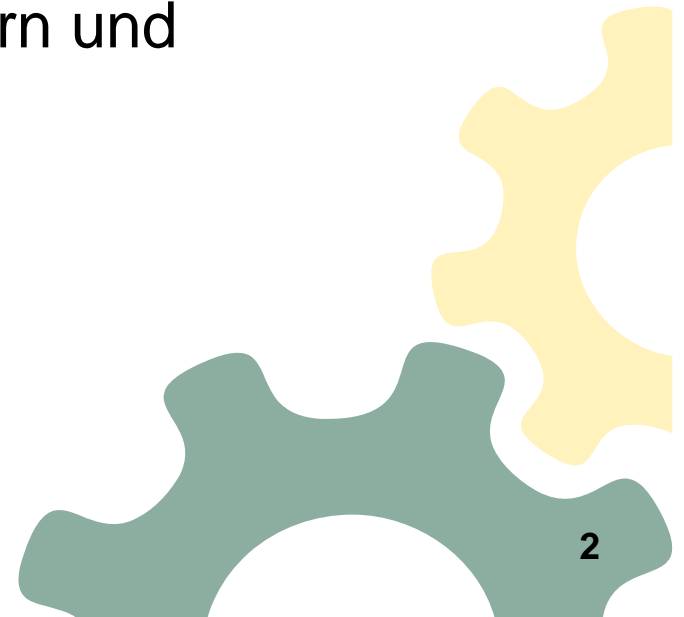
Lehrstuhl für Programmiersysteme

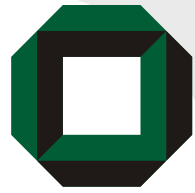




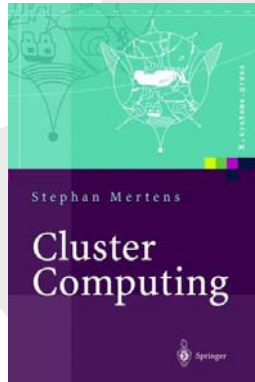
# Inhalt der Vorlesung Rechnerbündel

- Architektur von Multikernrechnern und Rechnerbündeln
  - Hochgeschwindigkeitsnetzwerke
  - Hochgeschwindigkeitskommunikation
- Betrieb von Rechnerbündeln
  - Administration
  - Ablaufplanung
  - Platzierung
- Programmierung von Multikernrechnern und Rechnerbündeln
  - BSP, MPI, JavaParty
  - DSM
  - OpenMP, Fäden
  - Parallele Algorithmen

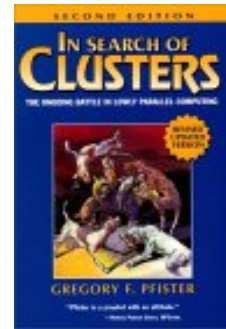




# Literatur



Cluster Computing  
Heiko Bauke, Stephan Mertens  
Springer Verlag, 2005  
ISBN: 3-540-42299-4



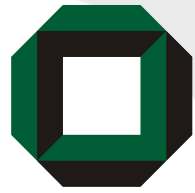
In Search of Clusters  
Gregory Pfister,  
Prentice Hall, 1998  
ISBN 0-13-899709-8



Parallele Programmierung,  
Thomas Rauber, Gudula Rünger,  
Springer Verlag, 2007  
ISBN 978-3540465492

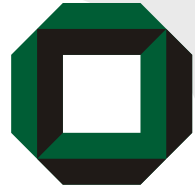


Parallel Programming in OpenMP,  
Rohit Chandra et al.,  
Morgan Kaufmann, 2000  
ISBN 978-1558606715



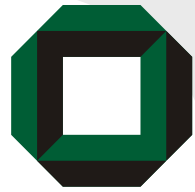
## Hilfe

- Vorlesungsfolien, Literaturhinweise:  
<http://www.ipd.uni-karlsruhe.de/Tichy/>
- Walter F. Tichy
  - Infobau, 3. OG, Zimmer 368
  - email: tichy @ipd.uni-karlsruhe.de
  - Tel: 608-3934, Sprechstunde Freitags, 13:00-14:00
- Victor Pankratius
  - Infobau, 3. OG, Zimmer 372
  - pankratius @ipd.uni-karlsruhe.de
  - Tel: 608-7333, Sprechstunde: über E-Post vereinbaren



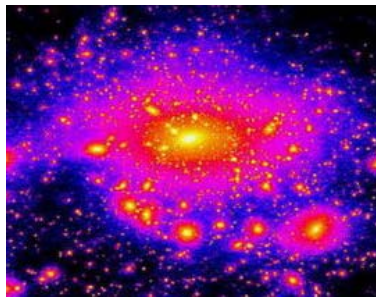
# Danksagung

- Die erste Rechnerbündel-Vorlesung entstand im WS 2001/02 an der Univ. Karlsruhe (Tichy und Moschny)
- Sie wurde ergänzt mit Beiträgen von
  - Prof. Dr. Philippsen, Uni Erlangen-Nürnberg
  - Prof. Dr. Christian Lengauer, Uni Passau
  - Prof. Dr. Michael Gerndt, TUM München
- Beiträge von Institutsmitarbeitern
  - F. Isaila, J. Reuter, B. Haumacher

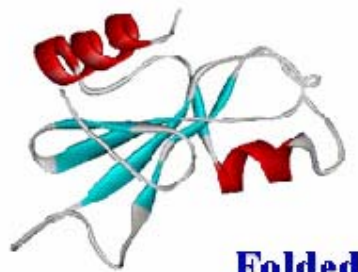


# Warum Parallelverarbeitung?

- Anwendungen brauchen mehr Leistung oder mehr Ressourcen als Ein-Prozessorsysteme bieten können.
- Warten auf nächste Hardware-Generation hilft nicht mehr → Taktfrequenzen steigen kaum noch.
- "Grand-Challenge" Applikationen aus Chemie, Astronomie, Bioinformatik, CAD/CAM, ...

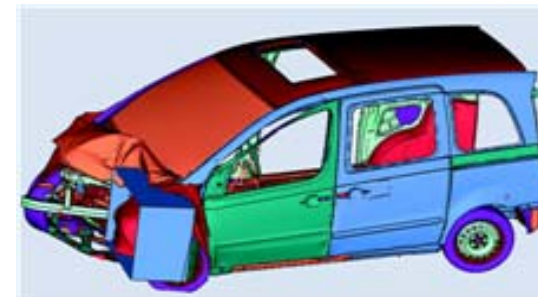


[physorg.com]



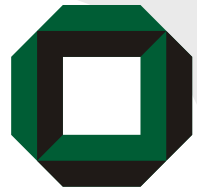
**Folded**

[IBM]



[Linux-Magazin.com]

- Simulationen, wenn Experimente unmöglich sind
- "Der Appetit kommt beim Essen"



# Ansätze zur Leistungssteigerung

Drei Arten, die Leistung zu steigern:

- **Härter arbeiten**

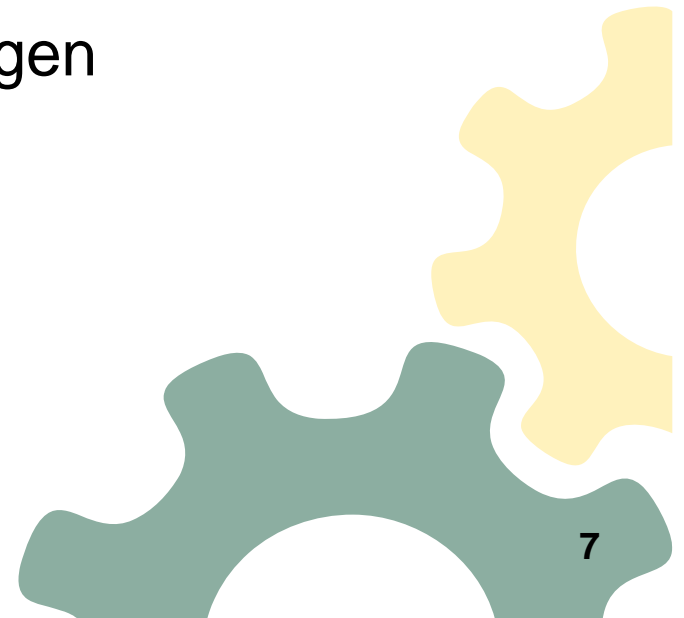
Schnellere Hardware verwenden,  
mehr Speicher einbauen, ...

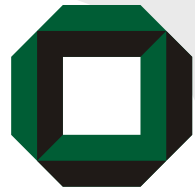
- **Schlauer arbeiten**

Effizientere Algorithmen, Optimierungen

- **Hilfe holen**

Parallelverarbeitung





# Ziele der Leistungssteigerung

Hätte man alle Ressourcen n-fach, ...

- **Durchsatz:**

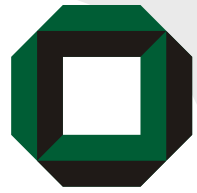
Berechne (hoffentlich) n Probleme simultan

- **Antwortzeit:**

Berechne 1 Problem in (hoffentlich) einem n-tel der Zeit

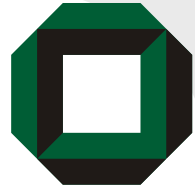
- **Problemgröße:**

Berechne 1 Problem mit (hoffentlich) einem n-fach vergrößertem Datensatz



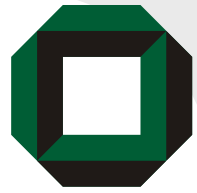
# Parallelität zur Durchsatzsteigerung

- Ausführung von  $n$  Instanzen eines *sequentiellen* Programms mit unterschiedlichen Datensätzen auf  $n > 1$  Prozessoren
- Z.B. Berechnung eines Films: Jeder Prozessor berechnet ein anderes Bild, verwendet aber dasselbe (sequentielle) Programm
- Problem: Ressourcen der einzelnen Prozessoren beschränkt
  - Begrenzte CPU-Leistung
  - Begrenzte Hauptspeichergröße
- Peinlich einfacher Parallelismus („embarassing parallelism“) tritt nicht häufig auf (z.B. Seti@home, Faktorisierung großer Zahlen, Datenanalyse der Hochenergiephysik)



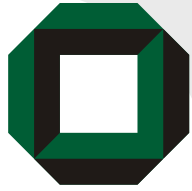
# Parallelität zur Beschleunigung

- Ausführung einer *einzelnen* Instanz eines *parallelen* Programms auf  $n > 1$  Prozessoren, wobei die CPUs dazu dienen, die Aufgabe gemeinsam schneller zu lösen.
- Z.B. Schnelles Darstellen von sehr großen (komplexen) Bildern: Jeder Knoten berechnet einen Teil des Bildes, um so die Programmausführung zu beschleunigen.
- Problem:
  - Paralleles Programm erforderlich
  - (Hohe) Anforderungen an die Kommunikationsleistung



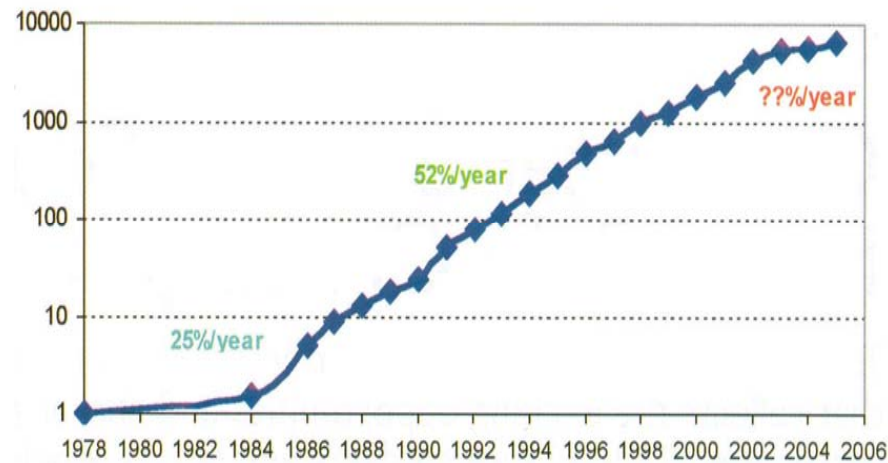
# Parallelität zur Problemgrößensteigerung

- Ausführung einer *einzelnen* Instanz eines *parallelen* Programms auf  $n > 1$  Prozessoren, wobei die Summe der lokalen Speicher dazu genutzt wird, größere Probleme zu berechnen.
- Z.B. Darstellen von sehr großen (komplexen) Bildern: Jeder Prozessor berechnet einen Teil des Bildes unter Verwendung seines lokalen Speichers.
- Problem:
  - Paralleles Programm erforderlich
  - (Geringe bis mittlere) Anforderungen an Kommunikationsleistung

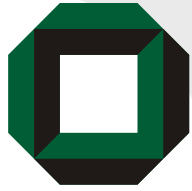


## David Patterson, Präsident ACM:

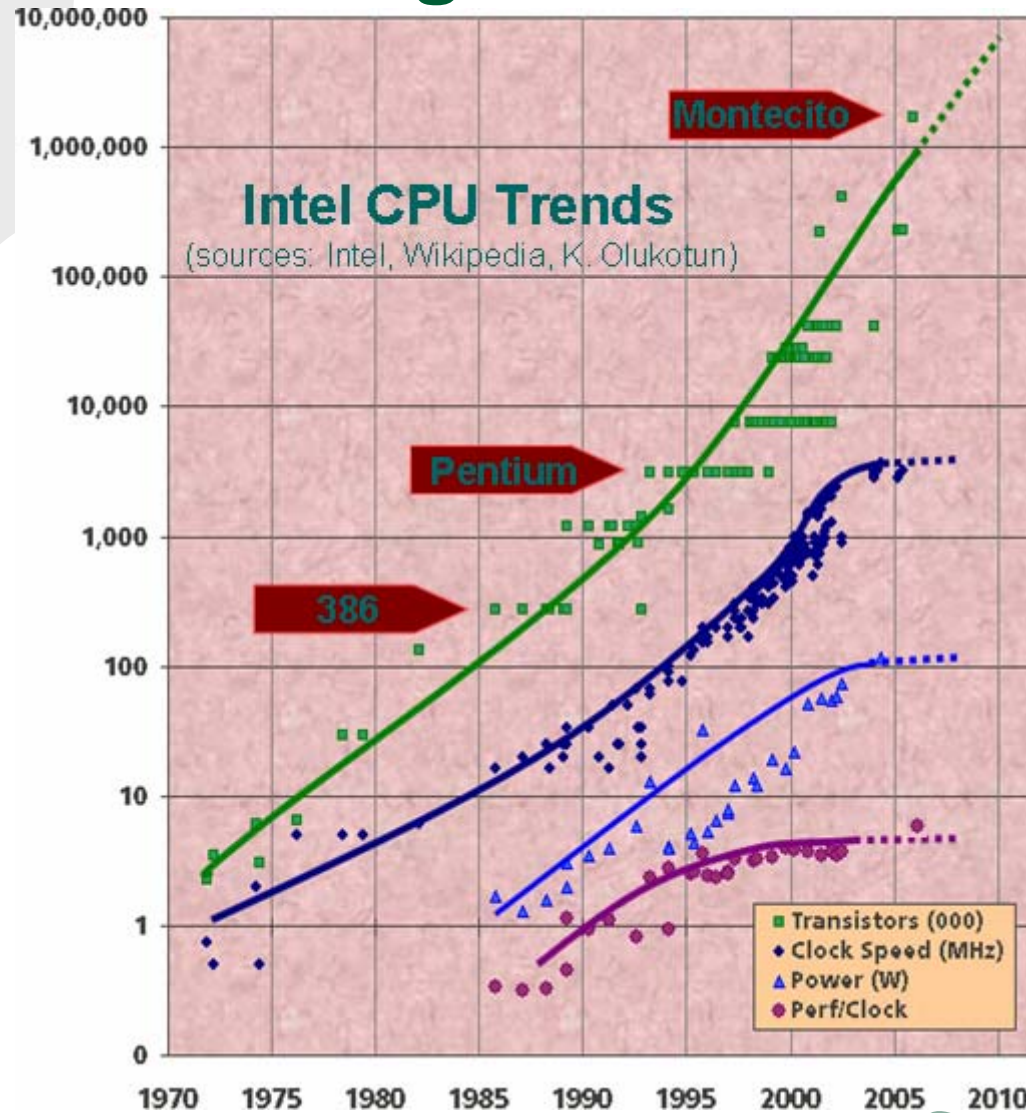
"In case you weren't paying attention, the era of doubling performance every 18 months ended in 2002."



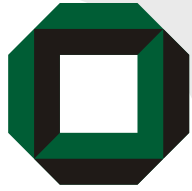
[CACM, April 2006]



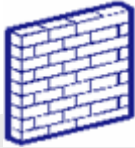
# Entwicklung der Rechnerleistung



Burton Smith,  
Manycore Computing  
Workshop, 2007

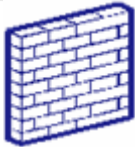


## Warum flachen die Kurven ab?



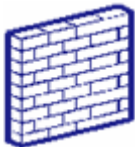
- „**Power Wall**“

Chips werden zu heiß (Energiedichte eines Nuklear-Brennstabes);



- „**Memory Wall**“

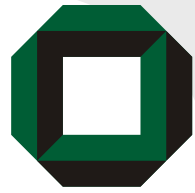
Keine ausreichende Verbesserung der Speicherlatenz.



- „**ILP Wall**“

Parallelismus auf Instruktionsebene ist ausgeschöpft;

➔ Die Zeit der "Gratis-Beschleunigung" geht zu Ende.

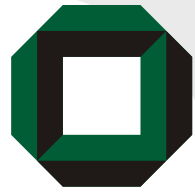


# Strategische Änderung: Multiprozessoren auf einem Chip

- Hersteller haben eine kritische Schwelle überschritten: Mehrere Prozessorkerne pro Chip sind möglich und preiswert:

Hersteller	#Kerne/Chip	etwa seit
IBM	2	2004
IBM	9	2005
AMD	2	2005
AMD	4	2007
Intel	2	2006
Intel	4	2007
SUN	8	2005

- Wachsende Anzahl von Prozessorkernen pro Chip sind prognostiziert (wegen weiterem Schrumpfen der Abmessungen der Transistoren, Leitungen, etc.)
- Intel stellt 2006 einen Prototypen mit 80 Prozessoren auf einem Chip vor



# Ihr Laptop— ein Parallelrechner?



**Inspiron™ 6400**  
15" Notebook für vielseitige  
Unterhaltung & 1 GB RAM.

~~729 €~~  
**659 €**  
inkl. MwSt., zzgl. 78 €  
Versand

**Prozessor** ?  
Intel® Pentium® **Dual-Core**  
T2080 Prozessor (1,73 GHz,  
533 MHz, 1 MB L2-Cache)



**Inspiron™ 1520**  
Stylischer Denker, der es  
genießt seine Qualitäten  
zeigen zu können und gerne  
im Mittelpunkt steht.

~~999 €~~  
**899 €**  
inkl. MwSt., zzgl. 78 €  
Versand

**Prozessor** ?  
Intel® Core™ 2 Duo T5450  
Prozessor (1,66 GHz, 667  
MHz, 2 MB L2-Cache)



**Inspiron™ 1720**  
Unterhaltung & Spaß  
garantiert! Technologie  
genau angepasst für Ihren  
Lifestyle. Jetzt in 8 Farben.

**1.049 €**  
inkl. MwSt. und Versand

**Prozessor** ?  
Intel® Core™ 2 Duo  
T5250 Prozessor (1,5  
GHz, 667 MHz, 2 MB  
L2-Cache)

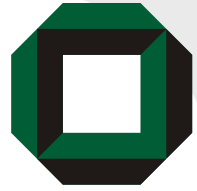


**Inspiron™ 1520**  
Schlank, elegant und noch  
etwas schlauer und stärker.  
Jetzt in 8 Farben.

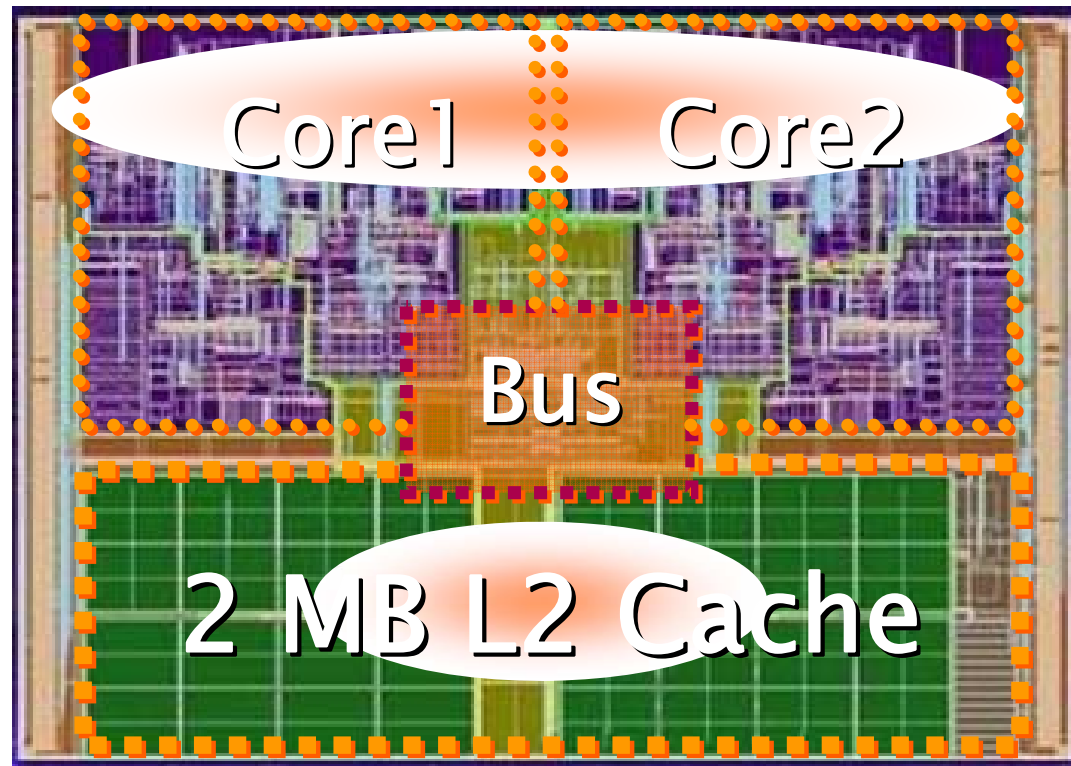
~~1.129 €~~  
**1.079 €**  
inkl. MwSt., zzgl. 78 €  
Versand

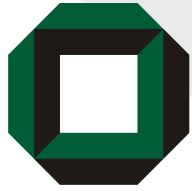
**Prozessor** ?  
Intel® Core™ 2 Duo T7100  
Prozessor (1,8 GHz, 800  
MHz, 2 MB L2-Cache)

Quelle: Dell 2007

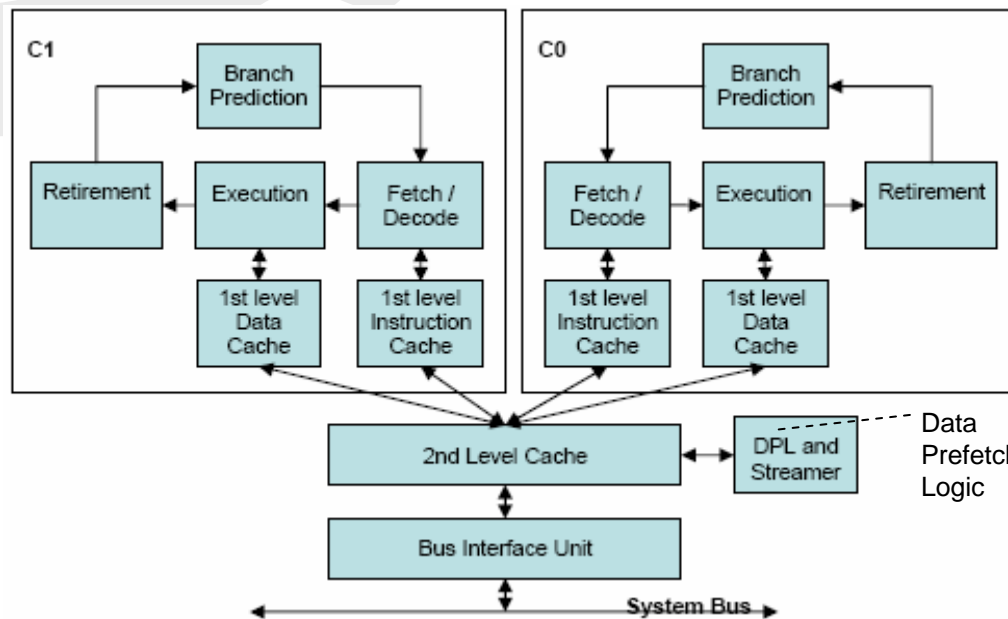


# Intel Core Duo Doppelprozessor

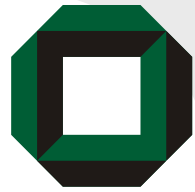




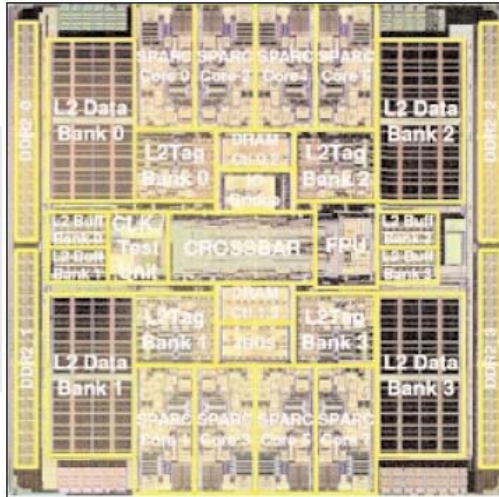
# Intel Doppelprozessor



Prozessoren: 2 Ghz  
32KB Datencache und 32 KB  
Instruktions-Cache pro Kern  
2MB gemeinsamer L2 Cache  
Bus: 667 MHz; 5,3 GB/s

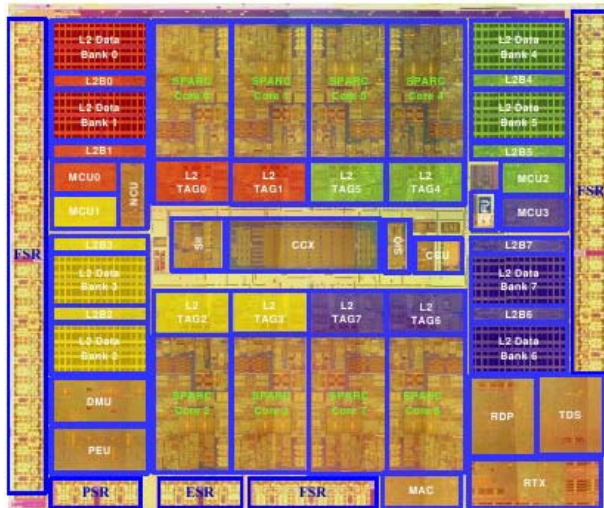


# SUN Niagara 8-Kerne Chip (1)



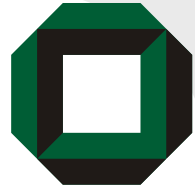
## Niagara 1

- 1.2 GHz
- ~ 279 Mio. Transistoren
- L1 Cache: 16KB (Instruction) + 8KB (Data) / Kern
- L2 Cache: 3MB (shared)
- 4 Threads / Kern
- 1 FPU



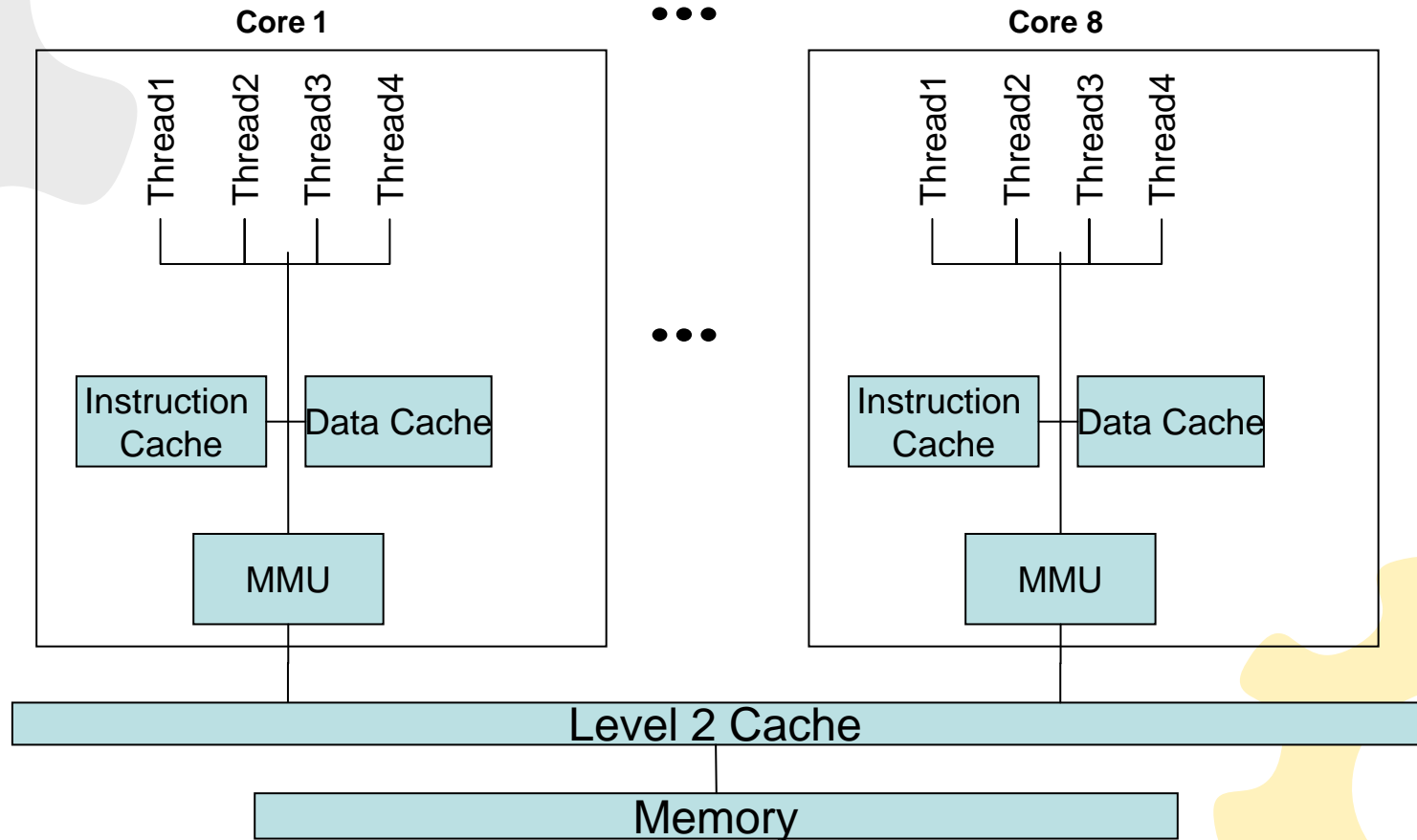
## Niagara 2

- 1.4 GHz
- ~ 500 Mio. Transistoren
- L1 Cache: 16KB (Instruction) + 8KB (Data) / Kern
- L2 Cache: 4MB (shared)
- 8 Threads / Kern
- 8 FPUs

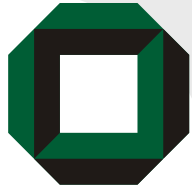


# SUN Niagara 8-Kerne Chip (2)

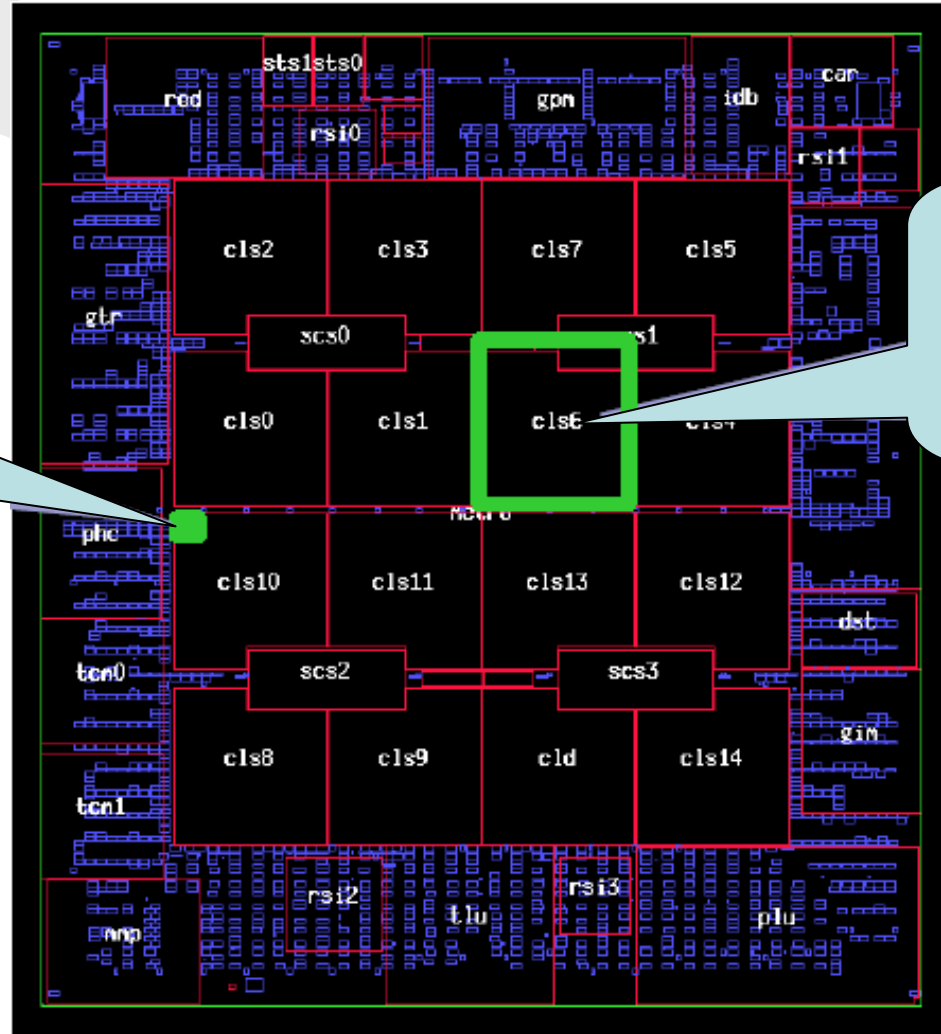
Niagara 1:  
4 Fäden/Kern



Niagara 2: analog mit 8 Fäden/Kern



# Cisco Metro: 192 Prozessoren auf 3,24 cm<sup>2</sup>

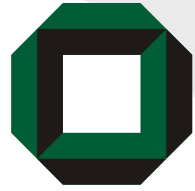


0,5 mm<sup>2</sup>  
pro  
Prozessor

16 Gruppen  
zu je  
12 Prozessoren  
(192 insgesamt)

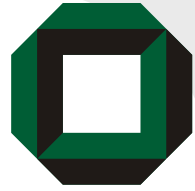
**192 Tenselica  
Prozessoren  
10 Gips  
250 MHz  
35W  
130nm Technik  
(2005)**

Bei aktueller  
45 nm Technik  
wäre Raum für  
das Achtfache an  
Prozessoren....



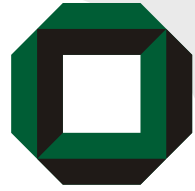
## Was sind die Folgen?

- Hauptproduktlinie der Prozessor-Hersteller sind Multikern-Chips.
- Server werden bereits seit 2005 mit Multikern-Chips ausgeliefert.
- Sogar Laptops werden ab 2006 mit Doppelprozessor-Chips ausgestattet (Dell, Apple, u.s.w.)



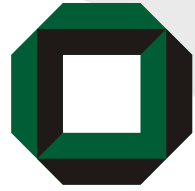
# Was ist hier neu?

- Multiprozessor-Betriebssysteme (für SMPs) sind gut entwickelt
  - Genügend Prozessparallelität, um auf Dienstgebern und selbst auf PCs moderate Multicore-Parallelität zu nutzen.
- Datenbanken und Transaktionsmanager haben schon lange mit Parallelismus gearbeitet.
- Web-Dienstgeber sind trivial parallel (viele parallele Fäden).
- Numerische Anwendungen sind oft bereits parallel.
- **Alltägliche Anwendungen sind nicht parallelisiert (z.B. Büroanwendungen, eingebettete Systeme, Spiele, etc.)**
  - Beherrschbarkeit der Parallelisierung für diese kritisch



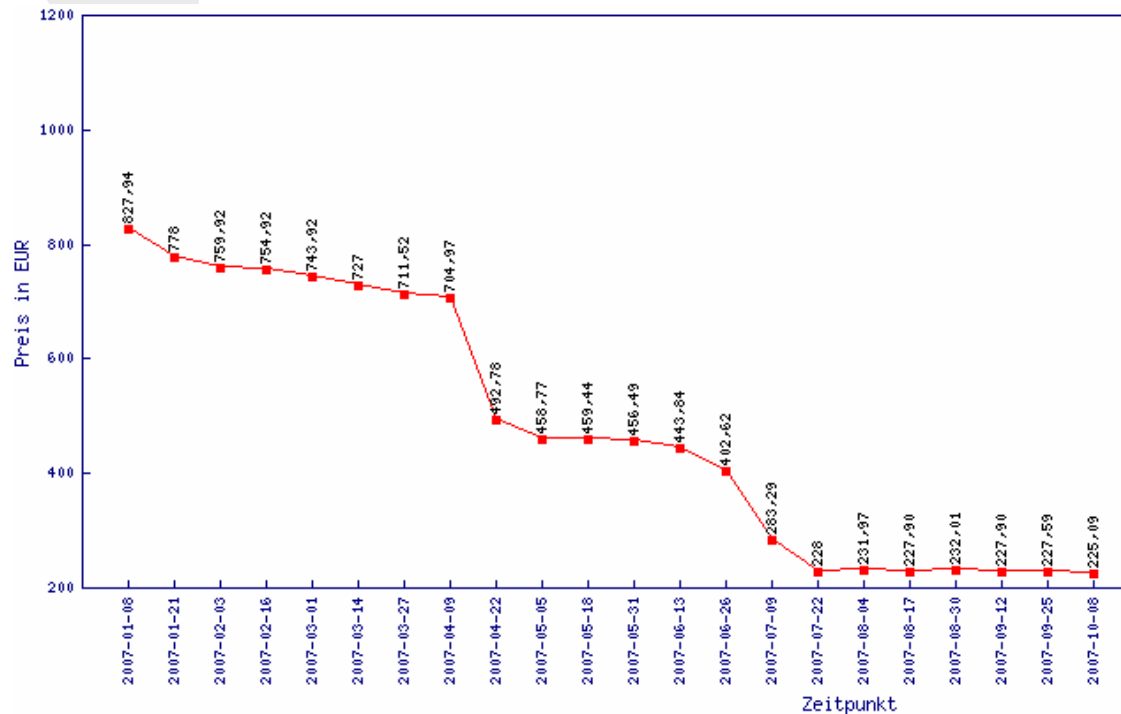
## Was sind die Folgen? (2)

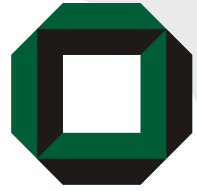
- Leistungssteigerungen von alltäglichen Anwendungen wird durch Parallelisierung erreicht.
- Parallelismus wird zum Normalfall.
- Informatiker müssen Parallelismus beherrschen lernen.
- **Fundamentaler Übergang vom sequentiellen zum parallelen Rechnen auf breiter Front mit dramatischen Auswirkungen auf Anwendungen, Forschung, und Lehre.**



## Was sind die Folgen? (3)

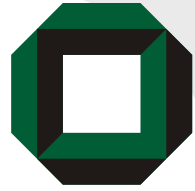
- Beispiel für Preisentwicklung für Intel Core 2 Quad Q6600 innerhalb eines Jahres





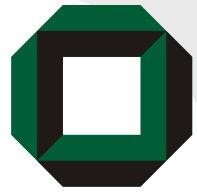
## Was sind die Folgen? (4)

- Die Softwarehersteller und die Informatikausbildung müssen rasch auf Parallelverarbeitung umstellen, um Wettbewerbsfähigkeit zu erhalten.
  - Umstellung von existierenden Anwendungen
  - Erstellung neuer, paralleler Anwendungen



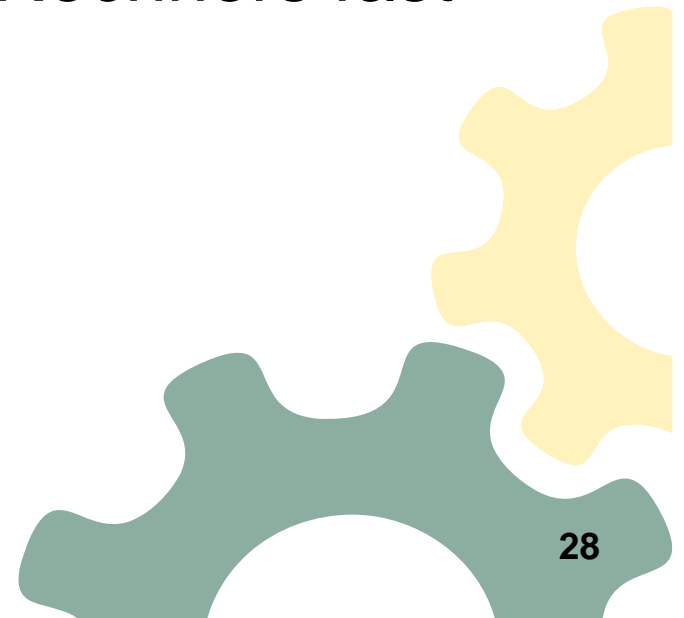
# Parallelität im Prozessor

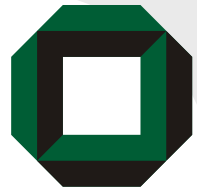
- Wortbreite
- Fließbandverarbeitung
- Superskalare Prozessoren
  - dynamische Ablaufplanung (Scheduling) für die Instruktionen (Datenabhängigkeiten zw. Instruktionen werden berücksichtigt)
- VLIW-Prozessoren
  - mehrere Instruktionen gleichzeitig,
  - statische Ablaufplanung
- Diese Parallelität wird implizit durch Übersetzer und den Prozessor ausgenutzt, kann aber nicht direkt programmiert werden.



## Und wenn das alles noch nicht reicht? Dann Bündel bilden!

- Rechnerbündel bestehen aus mehreren Einzelrechnern (evtl. aus SMP und Multikernrechnern) und einem Verbindungsnetz.
- Damit kann die jeweils verfügbare Spitzenleistung eines einzelnen Rechners fast beliebig vervielfältigt werden.





# Parallelrechnerklassifikation nach Flynn

globale Daten- u. Steuerungsflüsse als Kriterium

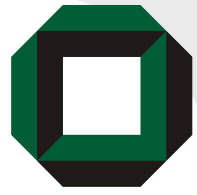
## SISD: Single Instruction, Single Data

- *eine* Verarbeitungseinheit, die Zugriff auf *einen* Datenspeicher und *einen* Programmspeicher hat.
- Klassischer sequentieller von-Neumann-Rechner



## MISD: Multiple Instruction, Single Data

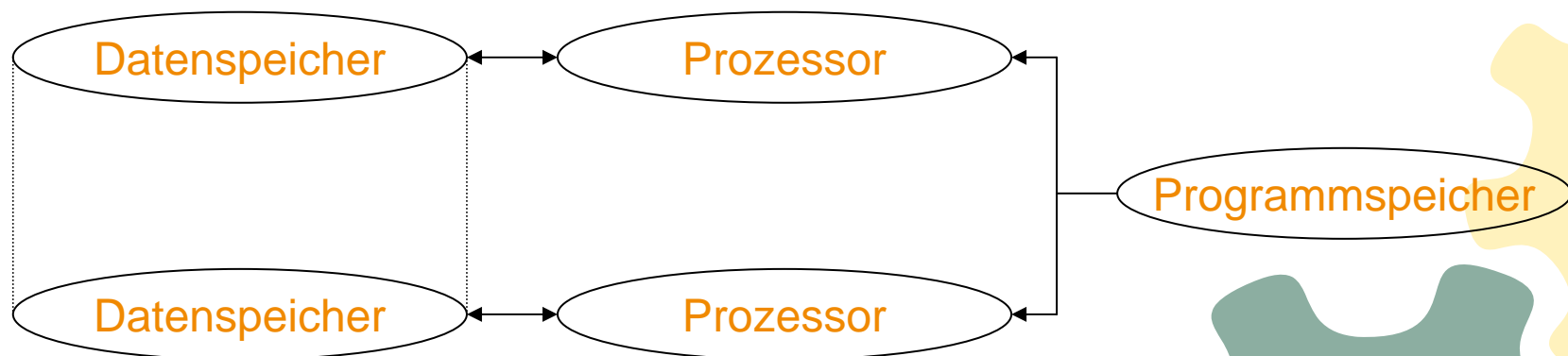
- Sog. Systolische Rechner: *ein* Datenstrom wird durch ein Fließband hindurch verarbeitet, wobei die Knoten im Fließband unterschiedliche Instruktionen ausführen.
- Keine praktische Bedeutung.

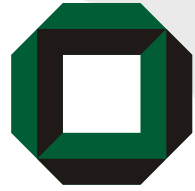


# Parallelrechnerklassifikation nach Flynn

## SIMD: Single Instruction, Multiple Data

- *Mehrere* Verarbeitungseinheiten; jede hat separaten Zugriff auf (gemeinsamen oder verteilten) Datenspeicher; ein Programmspeicher
- Synchrone Instruktionsabarbeitung





# SIMD-Beispiele

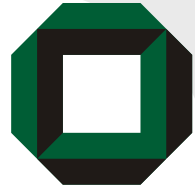
- Connection Machine CM-1, CM-2 von Thinking Machines
- MasPar MP-1, MP-2
- DAP, Distributed Array Processor

## Vorteile:

- Leichte Programmierung durch einen Kontrollfluss mit streng synchron-paralleler Abarbeitung aller Instruktionen

## Nachteile:

- Spezial-Hardware wurde vom Massenmarkt überholt
- Bedingte Anweisungen können zu ineffizienter Maschinennutzung führen



# SIMD: *if*-Anweisung

Anweisung1

```
if (lokale-Bedingung)  
  then {
```

then-Teil

```
} else {  
  else-Teil
```

```
}
```

Anweisung3

Alle Proz. führen Anweisung1 aus.

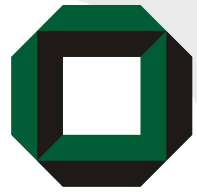
Alle Proz. werten Bedingung mit  
lokalen Daten aus.

Nur diejenigen Proz., bei denen  
lokale Bedingung erfüllt ist, führen  
then-Teil aus. Andere Proz.  
machen Pause.

Wechsel. Nun arbeiten die andere  
Prozessoren.

Synchronisationspunkt

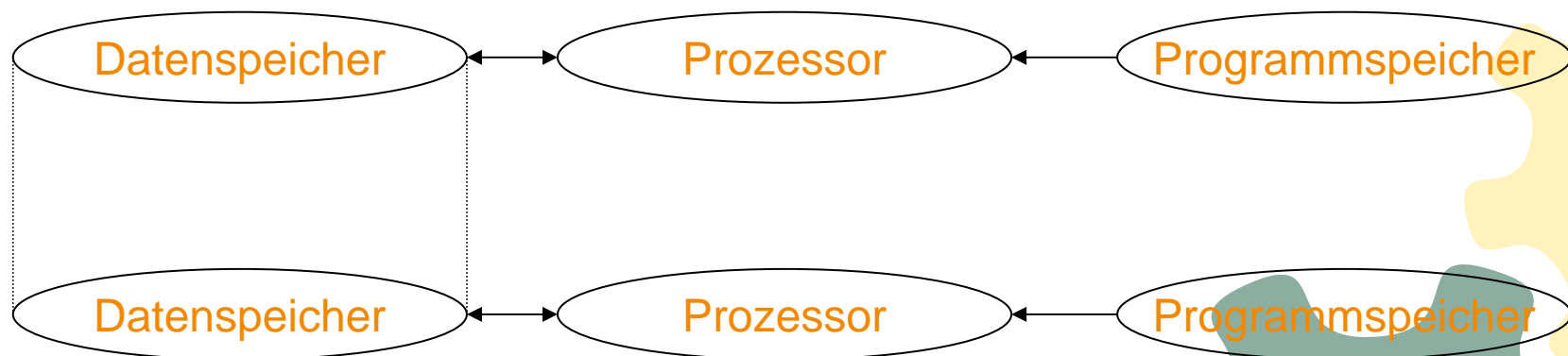
Alle Prozessoren führen Anweisung3  
aus.

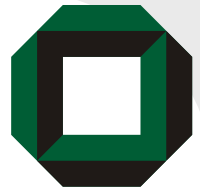


# Parallelrechnerklassifikation nach Flynn

## MIMD: Multiple Instruction, Multiple Data

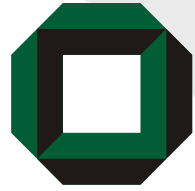
- *Mehrere* Verarbeitungseinheiten; jede hat separaten Zugriff auf (gemeinsamen oder verteilten) Datenspeicher; *mehrere* Programmspeicher





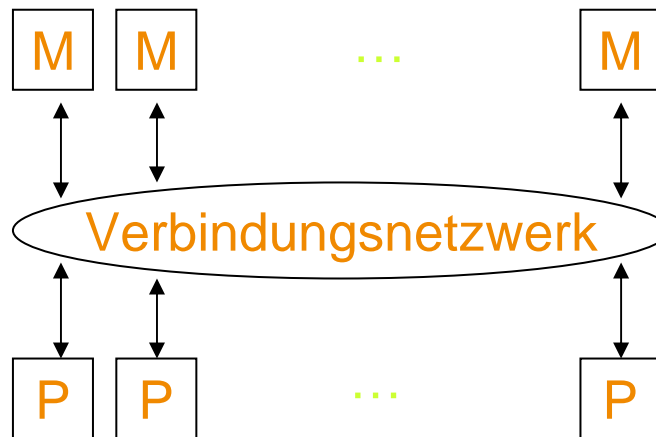
# Speicherorganisation v. Parallel-Rechnern

- Gliederung nach physikalischer Speicherorganisation
  - **Gemeinsamer Speicher/globaler Speicher**  
"Multi-Prozessorsystem"  
SMM – Shared Memory Machine
  - **Verteilter Speicher**  
"Multi-Computersystem"  
DMM – Distributed Memory Machine
- Gliederung nach Sichtweise für Programmierer
  - **Gemeinsamer Adressraum** (auf SMM und DMM)
  - **Verteilter Adressraum** (auf DMM)

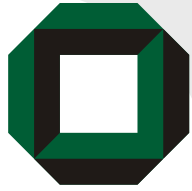


# Gemeinsamer Speicher

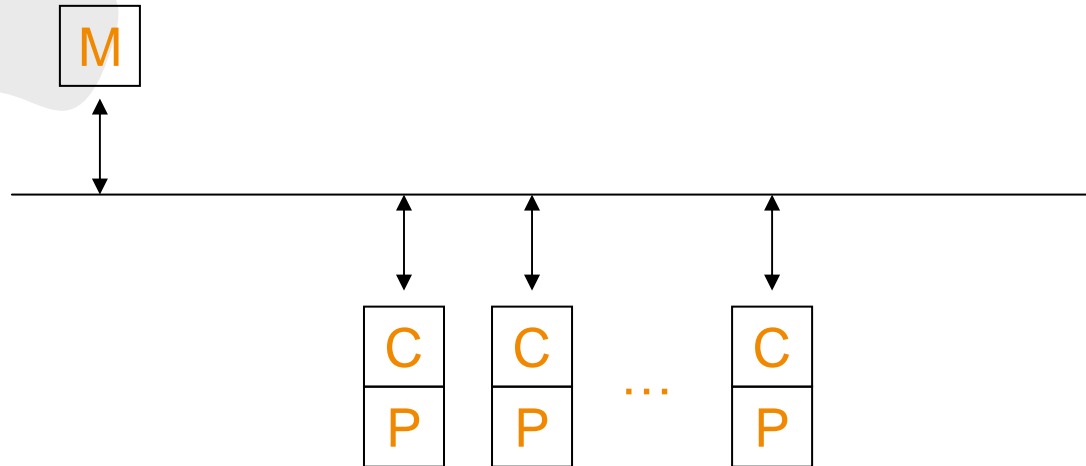
Konzept:



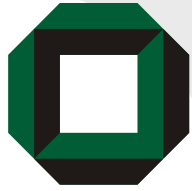
- Jeder Prozessor (P) kann über das Verbindungsnetzwerk direkt auf jeden Speicherbaustein (M) zugreifen.
- Das übliche Programmiermodell kann verwendet werden; kein explizites Senden und Empfangen von Nachrichten erforderlich
- Inkonsistenzen beim gleichzeitigen Schreibzugriff auf die selbe Speicherstelle (Schreibkonflikt) müssen in der Regel vom Programmierer verhindert werden.
- Unterscheidung: Zugriffszeit für "nahen"/"fernen" Speicher  
UMA oder NUMA: (Non-)Uniform Memory Access



# Gemeinsamer Speicher: SMP – Symmetric Multiprocessor



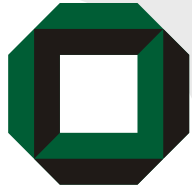
- Wenige Prozessoren, meist zentraler Bus, schlecht skalierbar
- Ein Adressraum, konzeptuell UMA – Uniform Memory Access
- In realen SMP ist aber ein lokaler Cache (C) pro Prozessor üblich
- Cache-Kohärenz in Hardware (Lesezugriff liefert immer den Wert des zeitlich letzten Schreibzugriffs. Später: Konsistenzmodelle)
- Ideal zur Programmierung mit parallelen Kontrollfäden (Threads)
- Bsp.: Multikernrechner, Doppelprozessorplatinen, SUN Enterprise, SGI Challenge



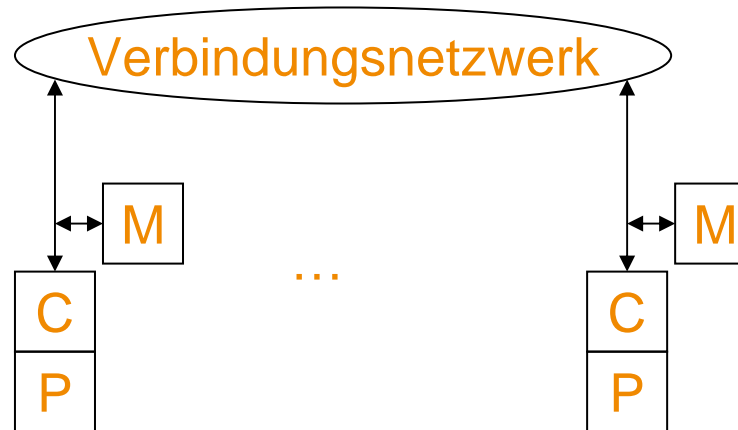
# Gemeinsamer Speicher: NUMA - Non Uniform Memory Access



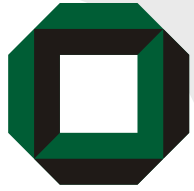
- Ein Adressraum, aber lokale Adressen sind schneller zu erreichen als entfernte Adressen
- Laufzeitunterschiede sind im Programm sichtbar
- Prozessoren sind schneller als Speicher, daher ist NUMA ohne Cache nicht sinnvoll.
- Wenn Cache vorhanden ist, dann nur für Daten aus dem lokalem Speicher eines Knotens.
- Bsp.: Cray T3D, T3E



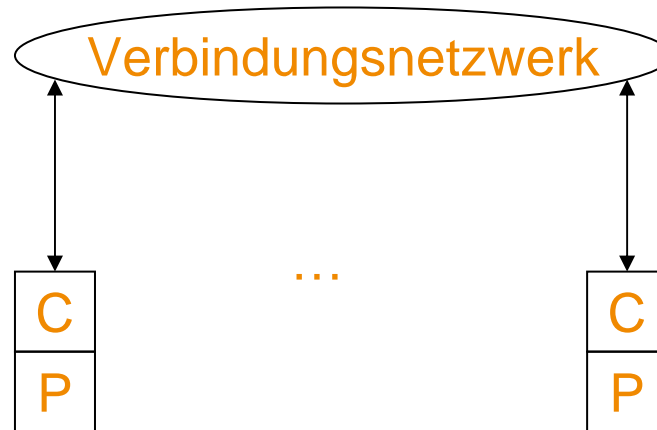
# Gemeinsamer Speicher: CC-NUMA – cache-coherent NUMA



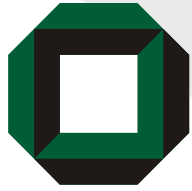
- Cache für lokale und entfernte Adressen
- Hardware sorgt für Cache-Kohärenz über den gesamten gemeinsamen Speicher.
- Skalierung problematisch, wegen Aktualisierungsdruck auf Caches (snoopy caching oder Rundruf von Schreibzugriffen)
- Bsp.: Stanford DASH, SGI Origin 2000



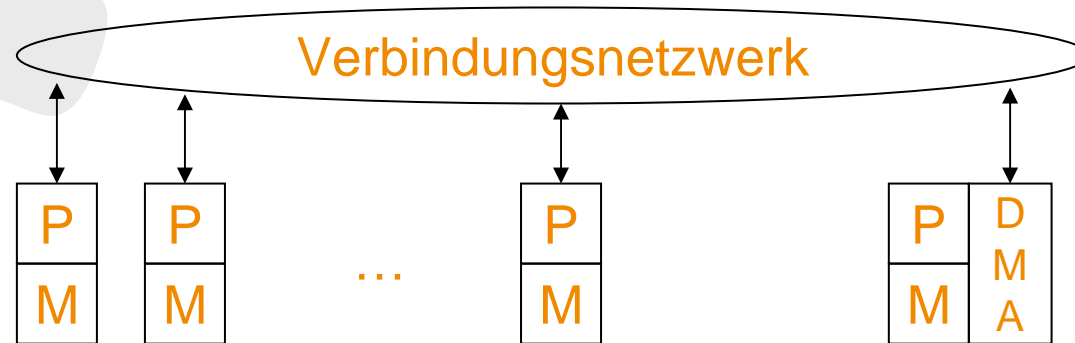
## Gemeinsamer Speicher: COMA – Cache Only Memory Access



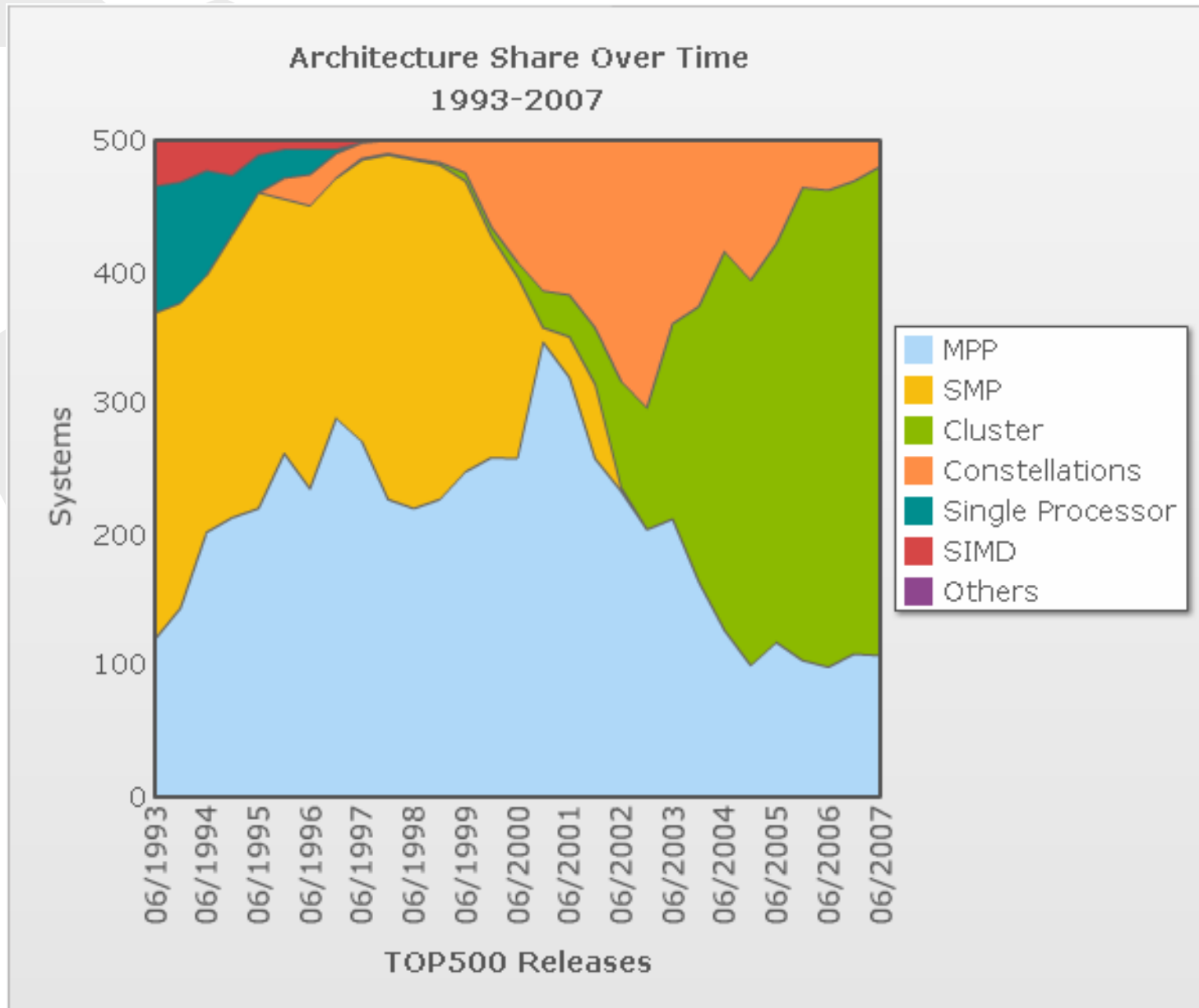
- Es gibt pro Knoten nur noch Cache-Speicher.
- Verteilte Cache-Speicher bilden gemeinsamen Speicher
- Hardware sorgt für Cache-Kohärenz
- Bsp.: Kendall Square Research KSR-1, KSR-2



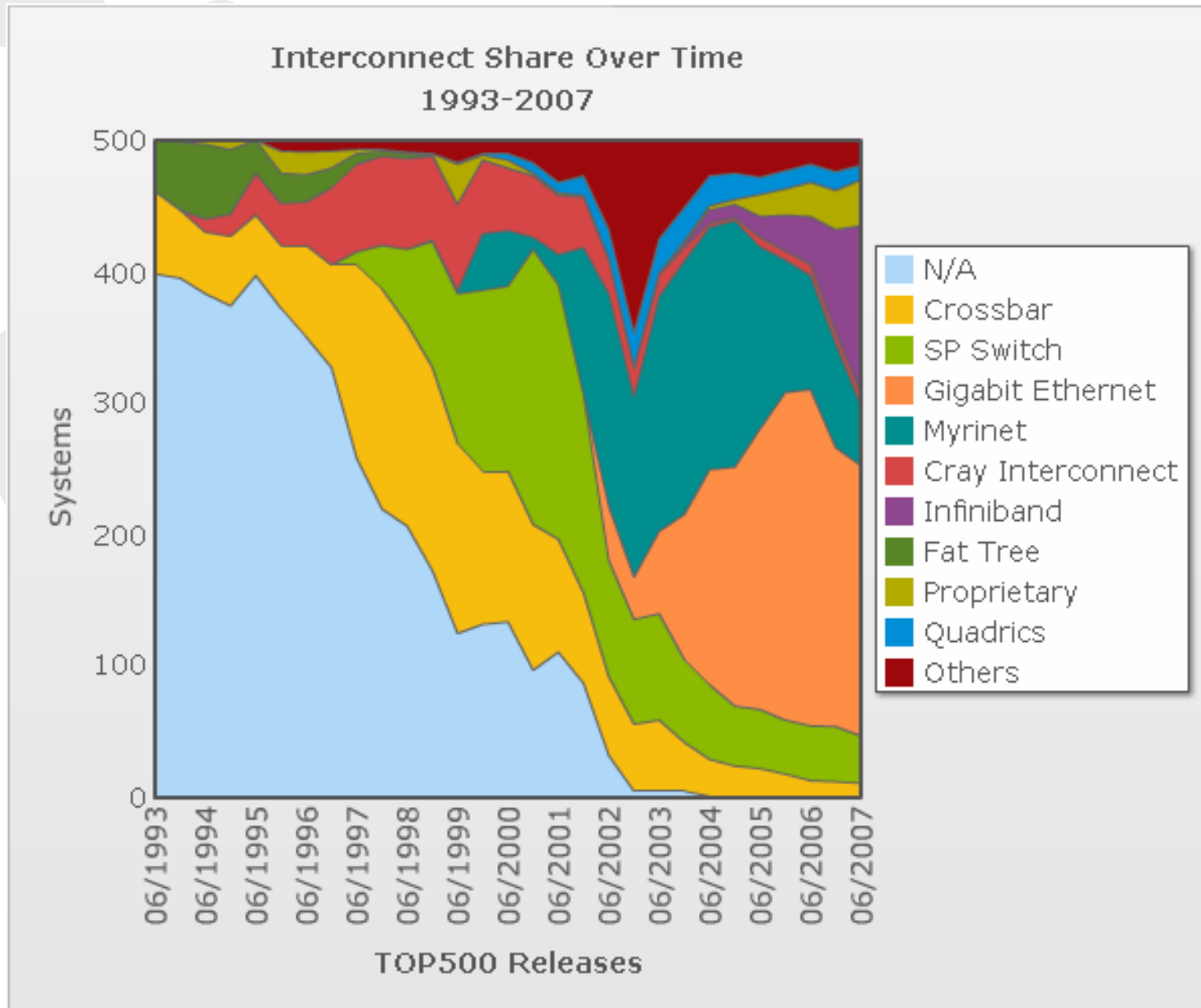
# Verteilter Speicher



- Nur der zugehöriger Prozessor (P) kann auf seinen privaten Speicher (M) direkt zugreifen.
- Zugriff auf entfernten Speicher nur über *explizit programmierten* Nachrichtenaustausch (komplementäre Sende- und Empfangsbefehle) möglich
- Lokaler Speicherzugriff (viel) schneller als Fernzugriff
- DMA (Direct Memory Access) für nebenläufigen Datentransfer zw. Speicher und Verbindungsnetz
- Bsp.: iPSC/860, CM-5, IBM SP2, ASCI Red/Blue/White etc.

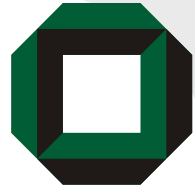


Quelle: [www.top500.org](http://www.top500.org)  
Stand: 06/2007



Quelle: [www.top500.org](http://www.top500.org)

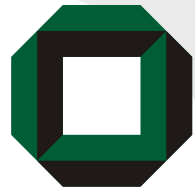
Stand: 06/2007



# Cluster/Rechnerbündel

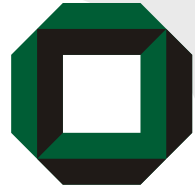
Rechnerbündel =

- Parallelrechner mit verteiltem Speicher (+ Verbindungsnetz)
- Knoten sind handelsübliche, vollwertige PCs oder Arbeitsplatzrechner (meist ohne Monitor, Tastatur, Maus, aber auch Multikern oder SMP-Knoten)
  - wegen Massenproduktion kosteneffizient
  - an vorderster Technologiefrent
  - nach verfügbarem Budget vergrößer-/modernisierbar
  - Stichwort COTS – Commodity of the Shelf
- Oft:
  - Linux auf jedem Knoten
  - Hochleistungsverbindungsnetz
- Preiswerte, leistungsfähige Alternative zu traditionellen Supercomputern



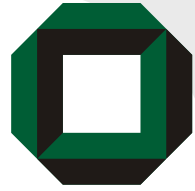
# Ursachen für Durchsetzung

- "Killer-Microprocessors"
  - Prozessorleistung verdoppelt sich alle 18-24 Monate
  - Die Rechner werden immer billiger (siehe Aldi-PC)
  - Standards machen Rechnertausch möglich
- "Killer-Networks"
  - Immer schnellere/breitere Netze (Myrinet, switched Gigabit-Ethernet, Infiniband)
  - Unbenutzte Zyklen auf Arbeitsplatzrechnern
- "Killer-Tools"
  - Programmierbibliotheken für die Massen, public domain

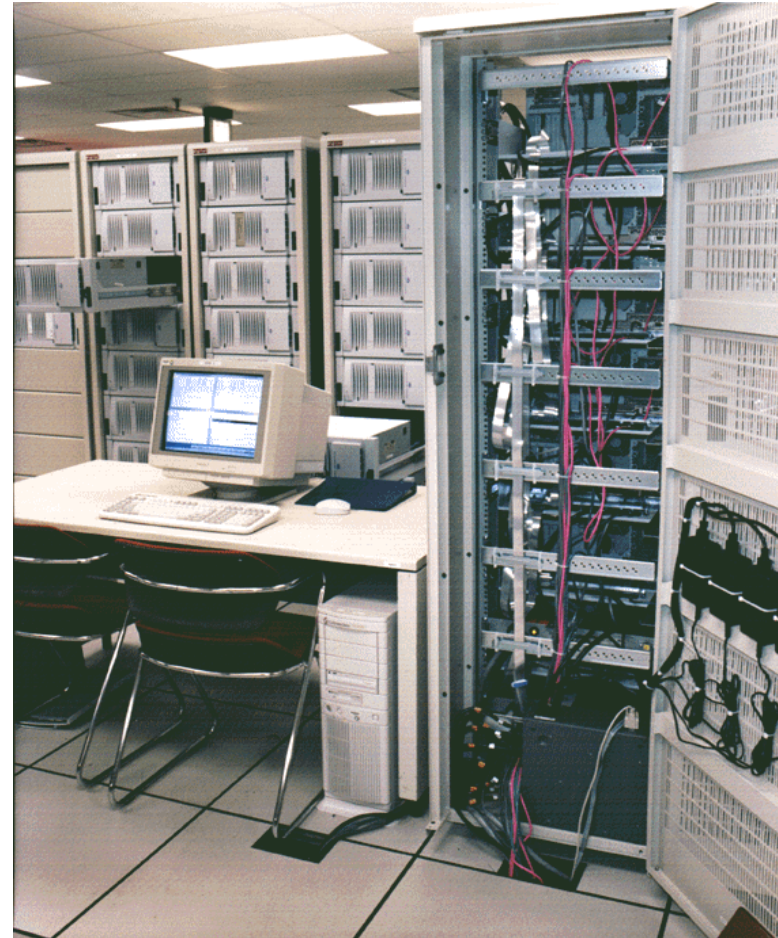


# Sorten von Rechnerbündeln

- Hochleistung versus Hochverfügbarkeit
  - Dediziertes Rechnerbündel versus Nicht-dediziertes
  - Knotentyp (Einprozessor-PC, SMP, Multikern-PC)
  - Homogen versus heterogen
  - Kommunikationsleistung: Bandbreite/Latenz
  - Ressourcen-Management
- 
- **Hier:** Hochleistung, dedizierte Bündel, homogene Systeme, hohe Kommunikationsleistung, Single-System-Image.



# Beispiele für Rechnerbündel (1)



CPLANT: Sandia National Lab, ca. 1350 Alpha 21264 (EV6 & EV67), Myrinet (2000)



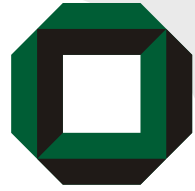
## Beispiele für Rechnerbündel (2)



*BEOWULF*: NASA Avalon Cluster  
140 Alpha 21164A, Fast Ethernet  
(1998)



Cluster im PC<sup>2</sup> Paderborn  
(200 x Dual Intel Xeon; 2005)



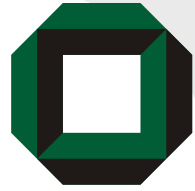
## Beispiele für Rechnerbündel (3)



AliCE Cluster Uni Wuppertal:  
128 Alpha 21264 (EV67), Myrinet  
Platz 210 auf Top 500 (11/2000)  
100 GFLOP, ParaStation



ITWM Cluster Kaiserslautern  
64 x Dual Intel Xeon, Myrinet  
Platz 296 auf Top 500 (06/2003)



## Beispiele für Rechnerbündel (4)

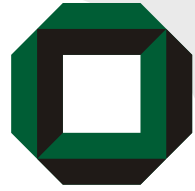


ALICENEXT: 11 TOWER, 43 NESTER



ZWEI TOWER MIT  
JE 48 KNOTEN

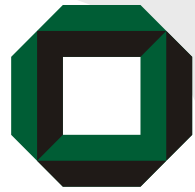
- AliceNEXT (Wuppertal)
- 3.7 TFLOP Spitzenleistung
- 2 TFlop Linpack
- Top500:
  - Platz 74 (06/2004)
  - Platz 167 (06/2005)
  - Platz 471 (06/2006)
- 512 Knoten (Superblades) mit je
  - 2 64-bit AMD Opteron Prozessoren (1.8GHz)
  - 2GB RAM
  - ~ 320GB Plattenplatz
  - 6x GBit Ethernet
    - 2on-board
    - PCI-X Quad GBitE Adapter
  - 32 Subcluster mit 2D-Torus (Mesh)
- 150 KW Leistungsaufnahme



# Top500 Top5: Blue Gene



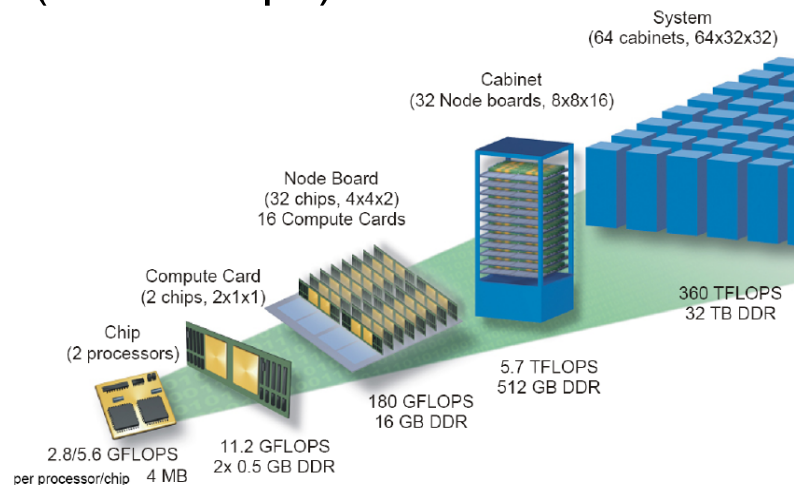
- **Platz 1** (BlueGene/L) der Top500 (Juni 2007)
- 131.072 PowerPC 440 (700Mhz)
- Linpack-Benchmark: 281TFlops
- Spitzenleistung 367TFlops
- Standort: Livermore (CA)
- Betreiber: DOE, NNSA, LLNL
- Betriebssystem: Linux



# Top500 Top5: Blue Gene

- Architektur

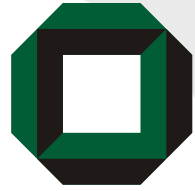
- Doppelkernprozessoren
- Karten mit 2 Chips
- Hauptplatinen mit 16 Karten
- Schrank mit 32 Hauptplatinen (1024 Chips)



- fünf voneinander

unabhängige Netzwerke

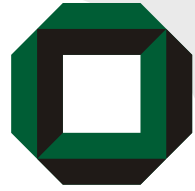
- Torus: P2P-Kommunikation
- "collective network" für Rundrufe und gemeinsam ausgeführte Arithmetikoperationen
- Barrierer-Netzwerk zur Synchronisation
- E/A mittels Gigabit-Ethernet-Netzwerk
- Kontrollnetzwerk zur Administration



# Top500 Top5: Jaguar



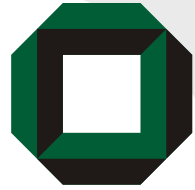
- **Platz 2** der Top500 (Juni 2007)
- 11.706 AMD dual-core Opteron (2.6 GHz)
- Linpack-Benchmark: 102TFlops
- Spitzenleistung 119TFlops
- Standort: Oak Ridge (TN)
- Betreiber: Oak Ridge National Laboratory
- Betriebssystem: Linux



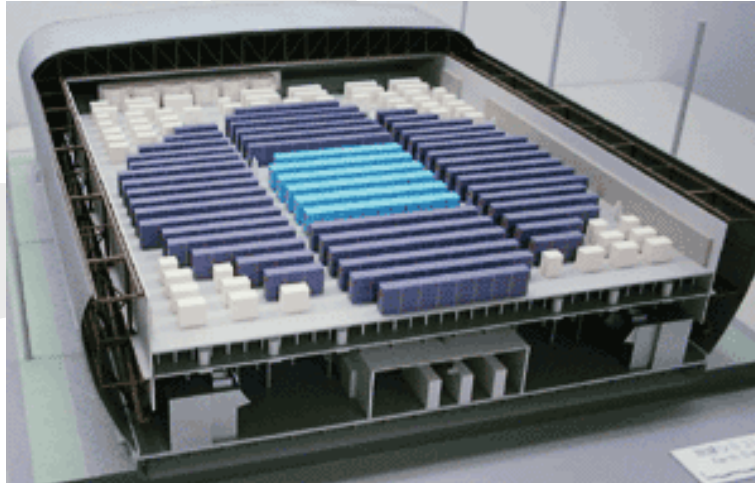
# Top500 Top5: Red Storm



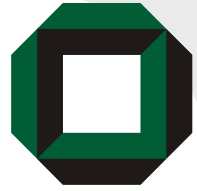
- **Platz 3** der Top500 (Juni 2007)
- 26.544 AMD dual-core Opteron (2.4 GHz)
- Linpack-Benchmark: 101TFlops
- Spitzenleistung 127TFlops
- Standort: Albuquerque (NM)
- Betreiber: NNSA/Sandia National Laboratories
- Betriebssystem: Linux



# Earth-Simulator



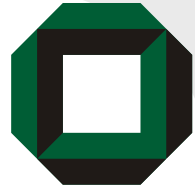
- Top500:
  - Platz 1 (06/2002 bis 06/2004)
  - Platz 4 (06/2005)
  - Platz 10 (06/2006)
  - Platz 20 (06/2007)
- Yokohama (Japan)
- 40TFLOPS Spitzenleistung
- 35,8 TFLOPS Linpack
- 5120 Arithmetik-Prozessoren, 500MHz – 1GHz, 8GF
- 640 Rechenknoten mit je
  - 8 Prozessoren
  - 16 GB RAM
- 640x640, 12.3GB/s x 2 Kreuzschienenverteiler
- 8TB/s gesamte Bandbreite
- Hersteller: NEC
- MPI / HPF
- Klimaforschung, Erdbebenforschung



# Carla: ParaStation Cluster

- Universität Karlsruhe,
- 32 Pentium III Prozessoren,  
800 MHz  
(16 Doppelprozessorknoten)
- Myrinet 2000
- Linux





# KIA: ParaStation Cluster

- Universität Karlsruhe,
- 32 Itanium 2 Prozessoren,  
1,3 GHz,  
(16 Doppelprozessorknoten)
- Infiniband
- Linux

