Einführung in das Hochleistungsrechnen

Martin Lanser

Universität zu Köln

SoSe 2019

Organisatorisches

Ablauf der Vorlesung und der Übungen

Termine

Vorlesung: Mo. 12:00 - 13:30 Uhr
Mi. 12:00 - 13:30 Uhr (außer 25.04.2019)
Ersatztermin: 01.04.2019, 14 Uhr, Seminarraum 1

Übung: Mi. 08:00 - 19:30 Uhr
Do. 14:00 - 15:30 Uhr
im Seminarraum 1 des Math. Instituts

Beginn regulärer Betrieb der Übungen:
Mi. 10.04.2019
Info: In der ersten Vorlesungswoche findet mittwochs in der Vorlesung eine Einführung in die Grundlagen der Programmiersprache C statt.

Ablauf der Vorlesung und der Übungen

Übungsblätter

Ausgabe: wöchentlich im Internet

Abgabe: Theorieaufgaben schriftlich im entsprechenden Kasten in Raum 3.01 des Mathematischen Instituts, Programmieraufgaben per E-Mail, weitere Hinweise auf der Homepage und in den Übungen

Tutorium

Mo. 14:00 - 15:30 Uhr im Seminarraum 1; erster Termin: 08.04.2019

Themen: Fragen zum Vorlesungsstoff; Fragen zur (parallelen) Programmierung; Hilfe bei Programmieraufgaben

Weitere Informationen auf der Homepage

`http://www.numerik.uni-koeln.de/18111.html`

Erwerb der Leistungspunkte

Klausur am Ende des Sommersemesters (Termin bald bekannt)

Zulassungsvoraussetzungen zur Klausur

- 50% erreichte Übungspunkte
- alle Programmieraufgaben erfolgreich bearbeitet
- Vorrechnen einer eigenen Lösung

Hinweis: Die Programmieraufgaben werden bepunktet, um ein entsprechendes Feedback zu geben, gehen aber nicht die Gesamtpunktzahl ein, d. h. sie sind von der 50% Grenze entkoppelt.

Und jetzt zu interessantereren Dingen...

Was ist HPC?

HPC (High Performance Computing) = Hochleistungsrechnen

Bereich des computergestützten Rechnens, der alle Rechenarbeiten umfasst, deren Bearbeitung einer hohen Rechenleistung oder Speicherkapazität bedarf.

Hochleistungsrechner = Supercomputer

Ein für das HPC geeigneter Computer.

http://rrzk.uni-koeln.de/cheops.html

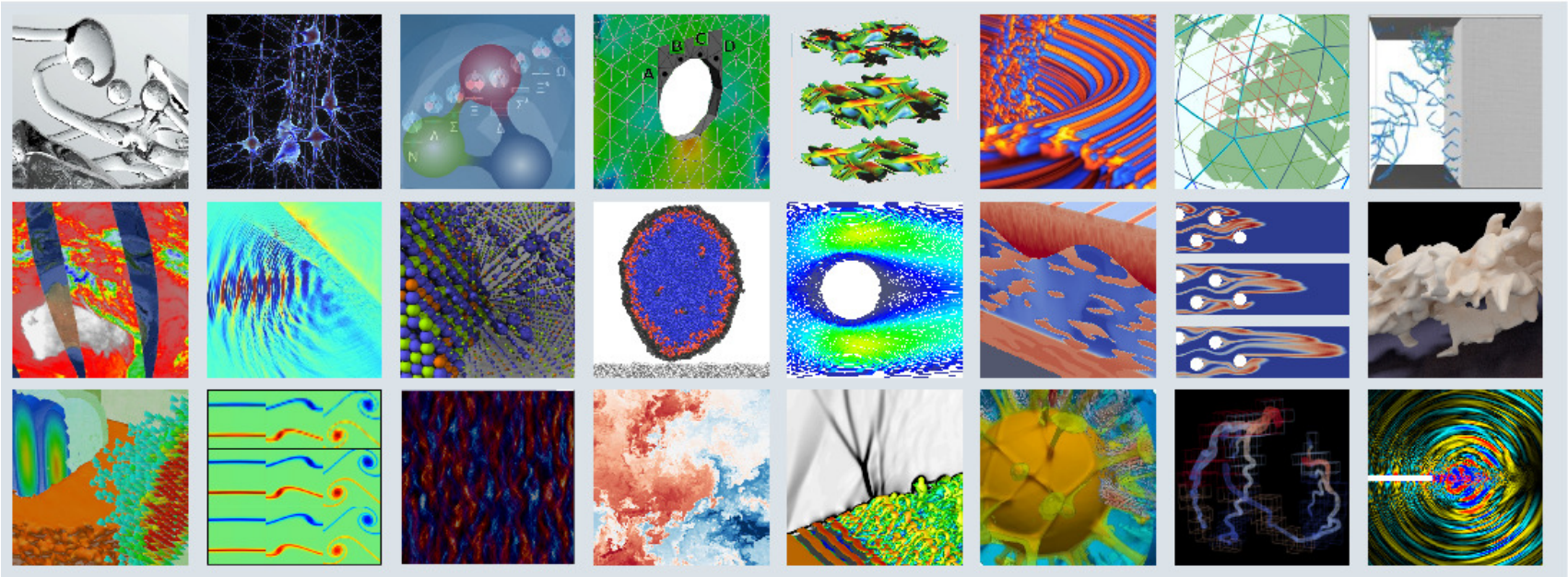
https://de.wikipedia.org/wiki/Hochleistungsrechnen

Warum HPC?

- Die Rechenleistung von Computern steigt immer weiter, aber auch der Bedarf an Rechenleistung.
- Immer komplexere Probleme und Simulationen interessieren Anwender und Wissenschaft:
  - menschliche DNA
  - komplexe und umfassende Klimamodelle
  - Simulation des Universums
  - medizinische Simulationen z.B. von Arterien, Herz, Gehirn usw.
  - die Verformung und Modellierung von Festkörpern
  - Forschung im Bereich der erneuerbaren Energie
  - Fluid Struktur Interaktion (FSI)

Large Scale Simulations (High-Q Club)

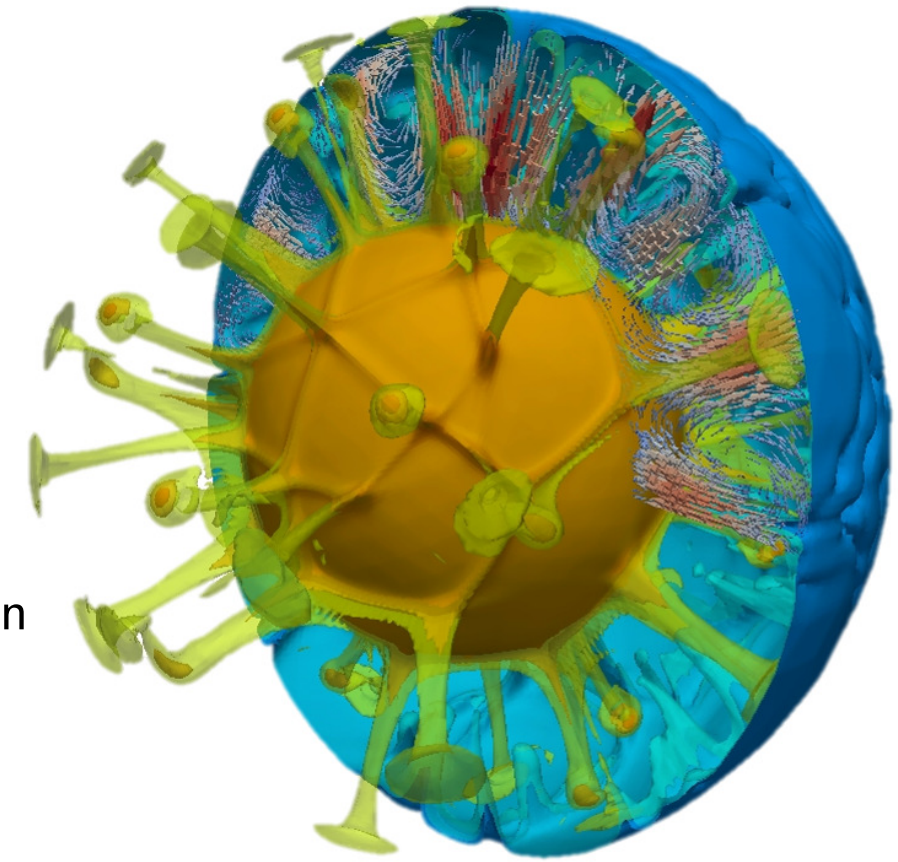
Die JUQUEEN, einer der größten Rechner Deutschlands, steht ganz in der Nähe im Forschungszentrum Jülich.
Einige große Simulationen aus verschiedenen Bereichen liefen auf der ganzen Maschine:



http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/_node.html

TERRA-NEO (High-Q Club)

Ein DFG gefördertes Projekt (SPPEXA), dass die Bewegungen innerhalb des Erdmantels simuliert.



Entwickler:

Ludwig-Maximilians-Universität München

Universität Erlangen-Nürnberg

Regionales Rechenzentrum Erlangen

Technische Universität München

http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/Terra-Neo/_node.html

ICON (High-Q Club)

Eine Software für Klimasimulationen mit z. B. hochauflösende Wolken. Dazu werden Icosaedergitter genutzt.

Entwickler:

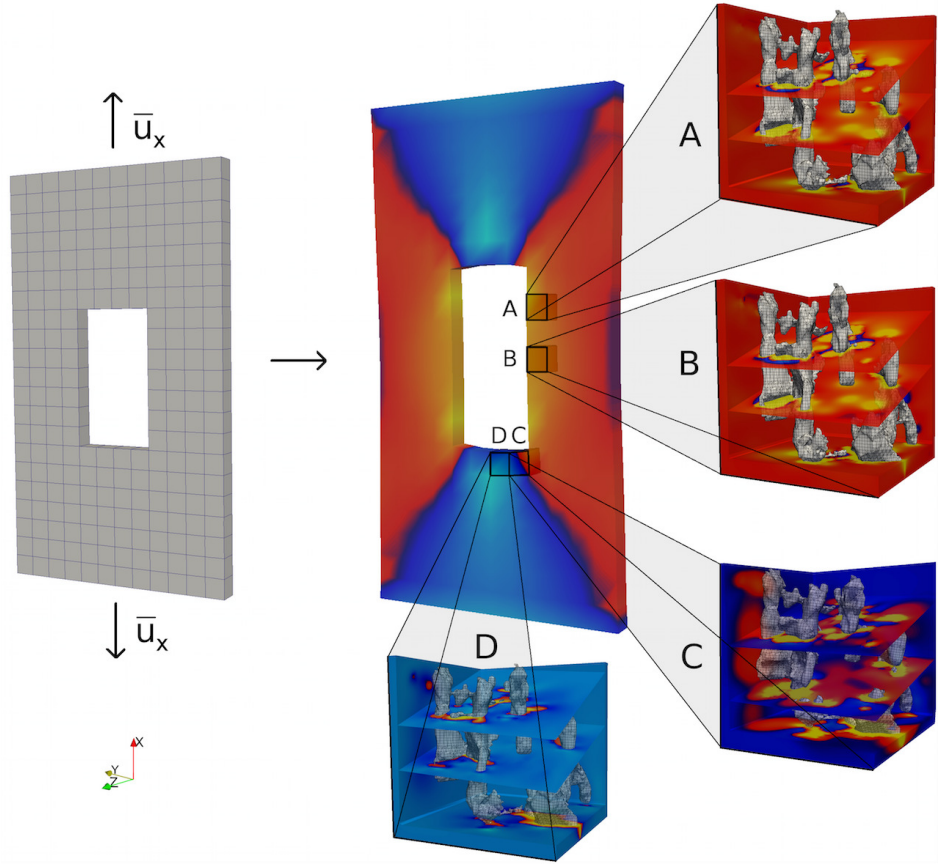
Max-Planck Institute for Meteorology Hamburg

Deutscher Wetterdienst Offenbach

http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/ICON/_node.html

FE2TI (High-Q Club)

Mehrskalen-Simulation von Stahlverformungsprozessen unter Berücksichtigung der mikroskopischen Struktur.



Entwickler:

Universität zu Köln

TU Bergakademie Freiberg

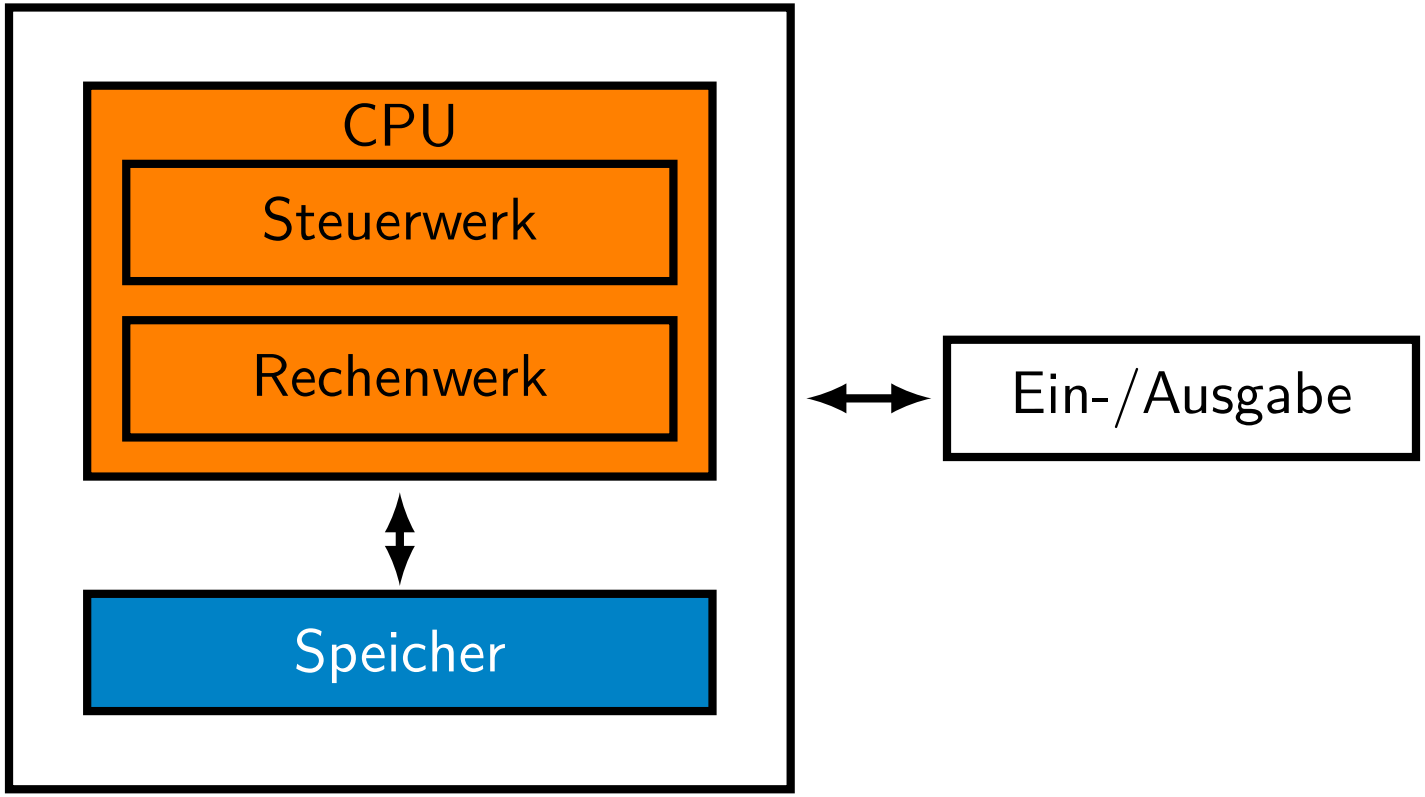
http://www.fz-juelich.de/ias/jsc/EN/Expertise/High-Q-Club/FE2TI/_node.html

Anforderungen an moderne Rechner

Was müssen moderne Rechner leisten um komplexe Simulationen zu ermöglichen? Was sind die Anforderungen solcher Simulationen?

- Viele Daten während der Rechnung $\leftrightarrow$ Viel Arbeitsspeicher (memory)
- Riesige Datenmengen an Ergebnissen $\leftrightarrow$ Große Festplatten und schneller Zugriff
- Möglichst kurze Rechenzeit $\leftrightarrow$ Schnellere Prozessoren oder mehr parallele Prozessoren

Von-Neumann-Rechner



CPU (Central Processing Unit) = (processor) core = (Prozessor-) Kern

Zentrale Verarbeitungseinheit eines Computers

Anforderungen an moderne Rechner

Was müssen moderne Rechner leisten um komplexe Simulationen zu ermöglichen? Was sind die Anforderungen solcher Simulationen?

- Viele Daten während der Rechnung $\leftrightarrow$ Viel Speicher (memory)
- Riesige Datenmengen an Ergebnissen $\leftrightarrow$ Große Festplatten und schneller Zugriff
- Möglichst kurze Rechenzeit $\leftrightarrow$ Schnellere Prozessoren oder mehr parallele Prozessoren

Anforderungen an moderne Rechner

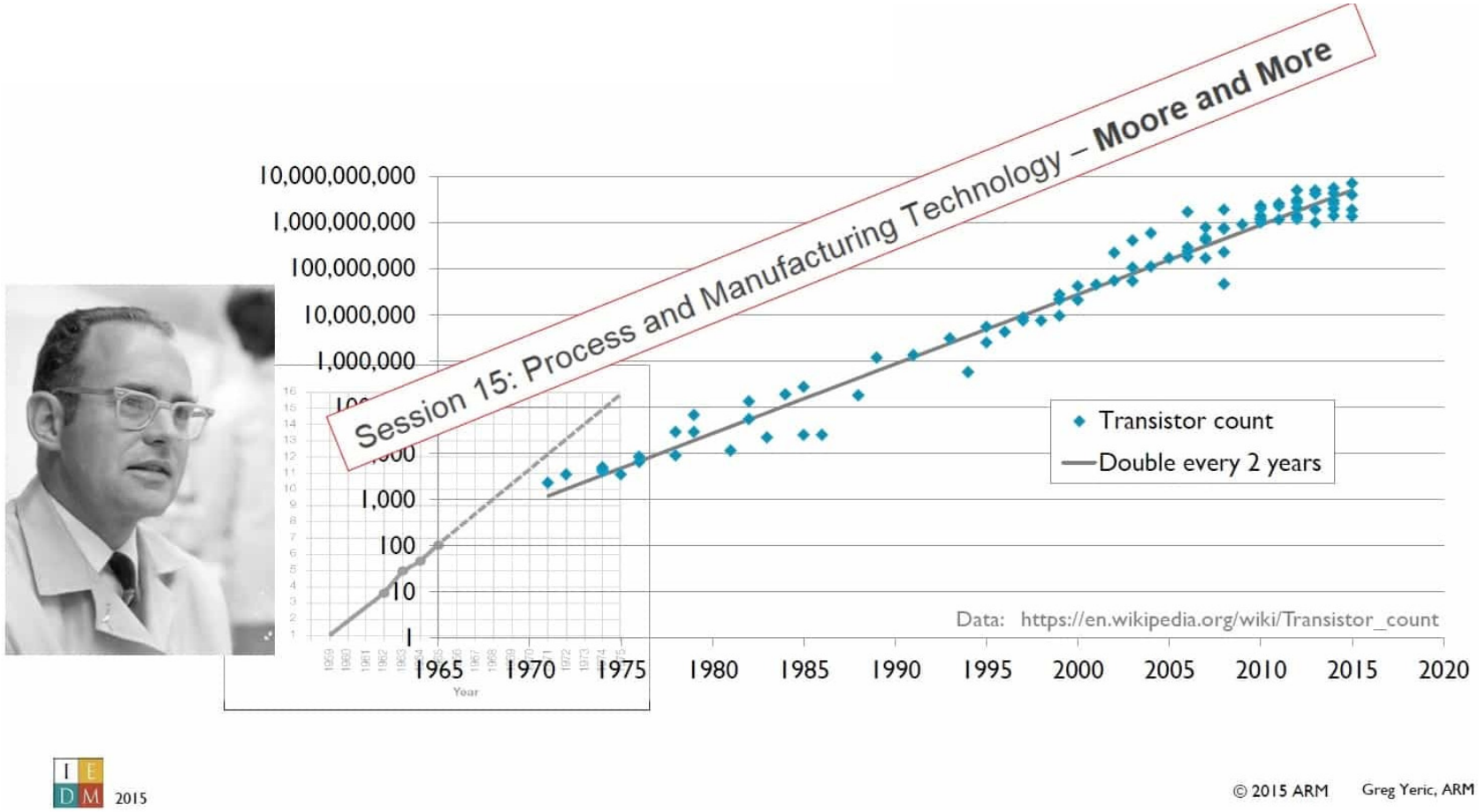
Was müssen moderne Rechner leisten um komplexe Simulationen zu ermöglichen? Was sind die Anforderungen solcher Simulationen?

- Viele Daten während der Rechnung $\leftrightarrow$ Viel Speicher (memory)
- Riesige Datenmengen an Ergebnissen $\leftrightarrow$ Große Festplatten und schneller Zugriff
- Möglichst kurze Rechenzeit $\leftrightarrow$ Schnellere Prozessoren oder mehr parallele Prozessoren

Moore's Law

- Bereits 1965 sprach Intel-Mitbegründer Gordon Moore von einer "jährlichen Verdopplung der Komplexität integrierter Schaltkreise".
- Damit ist gemeint, dass sich jährlich die Anzahl der Transistoren pro Flächeneinheit auf einem Computerchip verdoppelt, und das ohne gravierend steigende Produktionskosten.
- Im Klartext: Jedes Jahr eine Verdopplung der Rechenleistung (wobei nicht nur die Anzahl der Transistoren die Rechenleistung beeinflusst!)
- Diese Aussage ist auch als Moorsches Gesetz bzw. Moore's Law bekannt.
- Es stellte sich heraus, dass eine jährliche Verdopplung etwas hoch gegriffen war und sich eher alle 18-24 Monate die Anzahl an Transistoren eines Prozessors verdoppelte.

Moore's Law



https://www.semiwiki.com/forum/attachments/content/attachments/

16313d1452814173-arm-2015-moores-law-min-jpg

Moore's Law

- Dem Moorschen Gesetz wird (zumindest im Wortsinn) ein nahes Ende vorhergesagt, denn es gibt physikalische Grenzen:
  - Ein Transistor kann niemals kleiner sein als wenige Atome. Die Industrie ist nur noch wenige Prozessorgenerationen davon entfernt.
  - Ein Signal kann nicht schneller als mit Lichtgeschwindigkeit gesendet werden.
- Zudem gibt es viele akute Probleme:
  - Höhere Taktfrequenz $\rightarrow$ Mehr Energieverbrauch
  - Mehr Energieverbrauch $\rightarrow$ Höhere Temperatur
  - Höhere Temperatur $\rightarrow$ Unzuverlässigere Prozessoren $\rightarrow$ Mehr Kühlung
  - Mehr Kühlung $\rightarrow$ Noch mehr Energieverbrauch

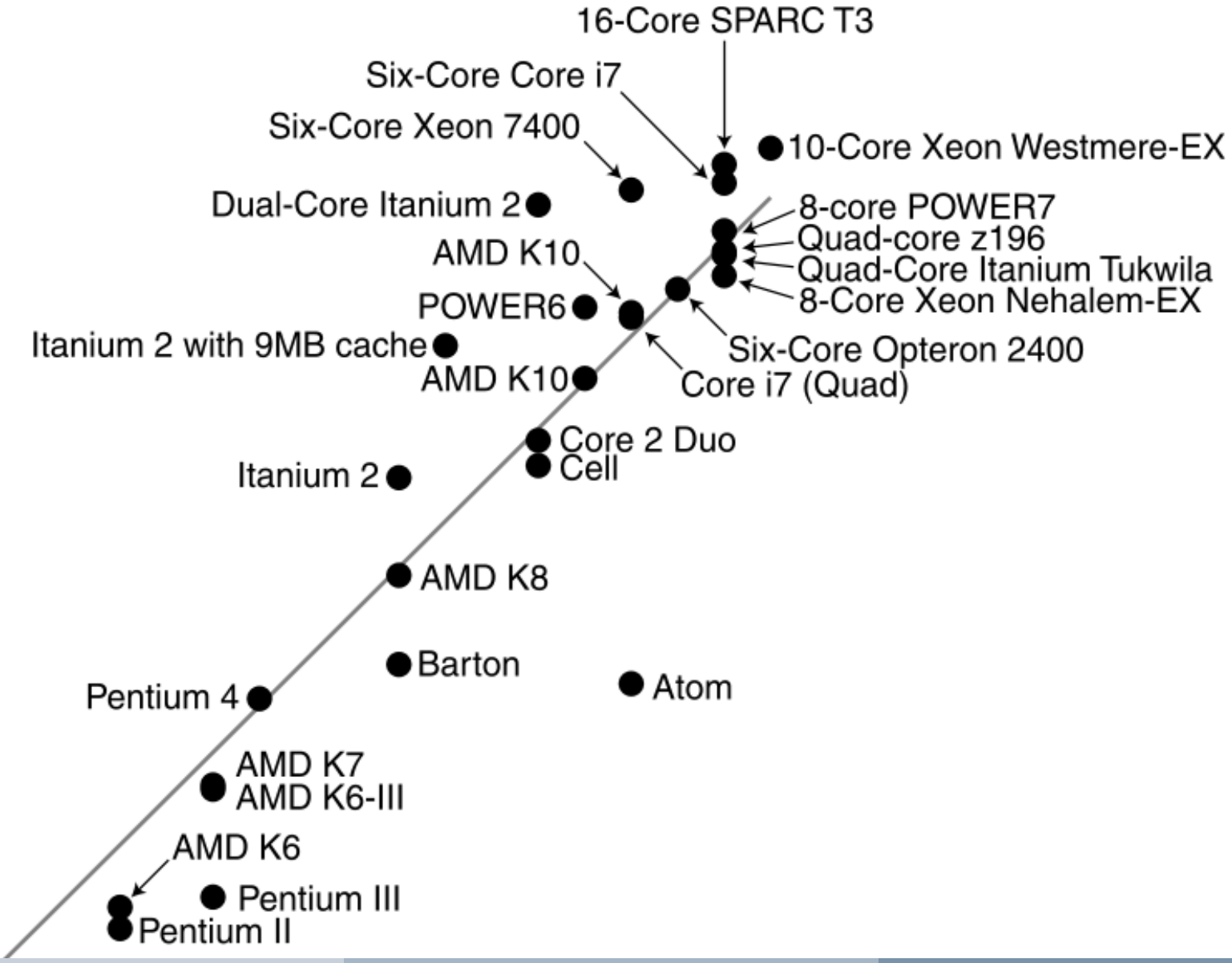
Moore's Law

- Tatsächlich steigt die Rechengeschwindigkeit eines Prozessorkerns bereits seit 2002 nicht mehr um 50% pro Jahr, sondern lediglich um etwa 20%.
- Dies widerspricht eigentlich der soeben gezeigten Grafik. Dort sah es aus, als würde Moore's Law noch immer gelten.
- Dies liegt an sogenannten Mehrkernprozessoren.
- Auch in Laptops oder Heimrechnern sitzen zumindest zwei CPUs auf einem Chip.

Mehrkernprozessor = multi-core processor

Prozessor mit mehr als einem Prozessorkern (core, CPU).

Moore's Law - Seit 1997



Moderne Supercomputer – Terminologie

Parallelrechner

Ein Hochleistungsrechner mit $p > 1$ Prozessoren, die *gleichzeitig* an einer Aufgabe arbeiten.

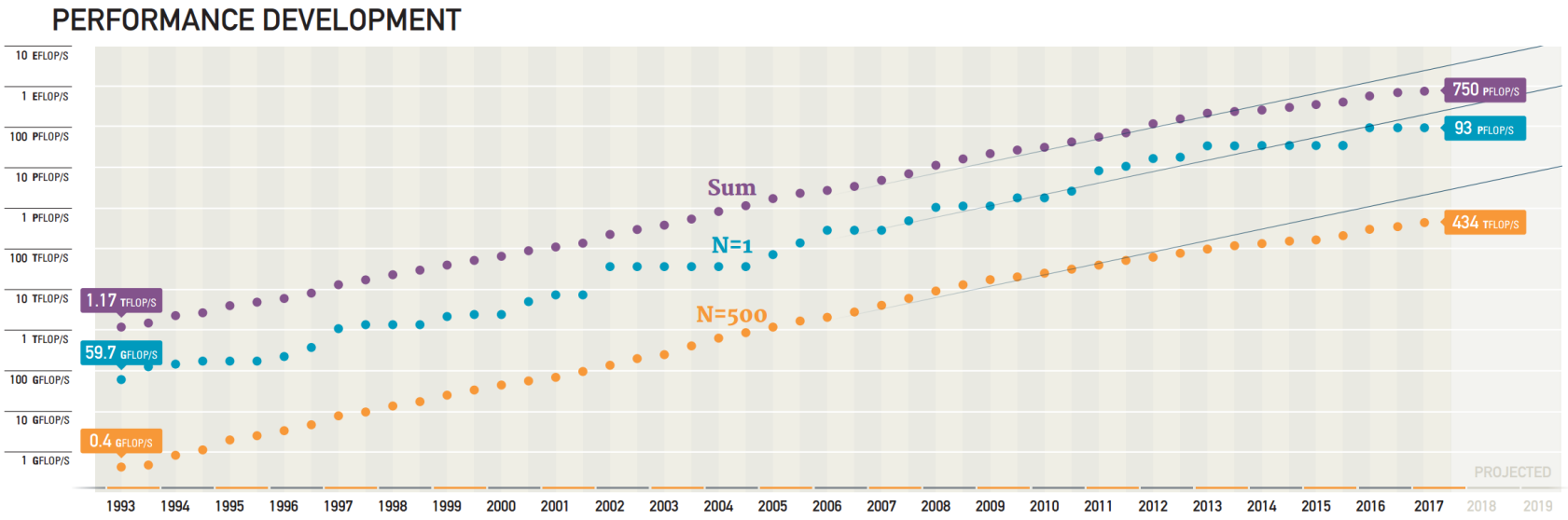
FLOPS = Floating Point Operations Per Second

Ein Maß für die Leistungsfähigkeit eines Computers: Anzahl von Gleitkommaoperationen pro Sekunde
1 TFLOPS $= 10^{12}$ FLOPS
1 PFLOPS $= 10^{15}$ FLOPS

TOP 500 (Stand: Juni 2017)

Liste der 500 schnellsten Rechner der Welt unter `www.top500.org`.

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
3	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 Cray Inc.	361,760	19,590.0	25,326.3	2,272
4	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209

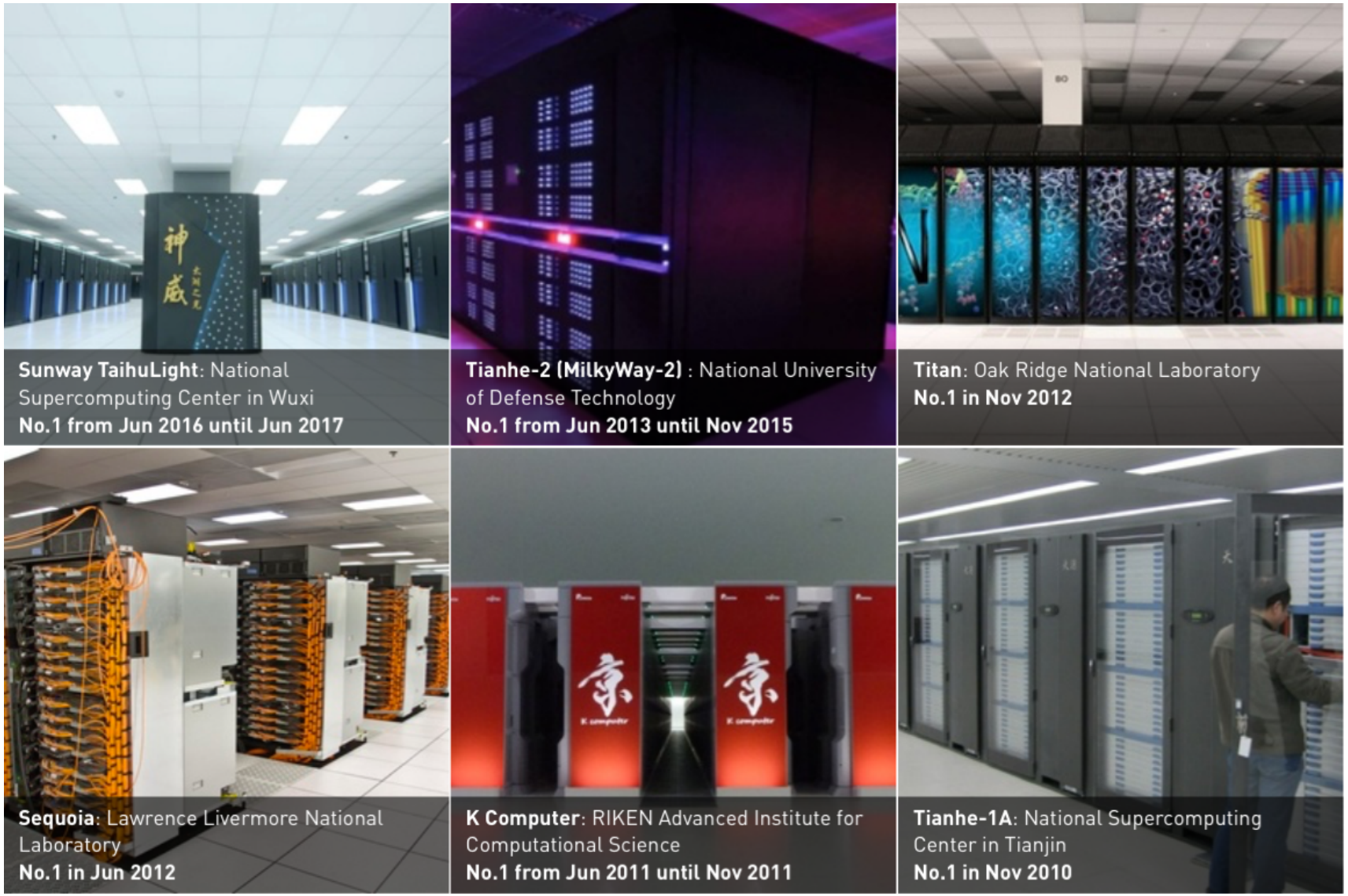


Europas größte Rechner (Stand: Juni 2017)

Rang TOP500	Name	Ort	#Cores	Peak-P. TFlops/s
3.	Piz Daint	CSCS, Schweiz	361.760	25.326,3
13.	MareNostrum	SC Center, Barcelona	148.176	9.957,4
14.	Marconi Intel Xeon Phi	CINECA, Bologna	241.808	10.833,0
17.	Hazel Hen	HLRS, Stuttgart	185.088	7.403,5
–	CHEOPS	RRZK Köln	9.712	100

Anmerkung: Im November 2010 startete CHEOPS auf Platz 90 in die TOP500 .

Die Nummer 1 im Laufe der Jahre



Green 500

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	7,031.58	Institute of Physical and Chemical Research (RIKEN)	Shoubu - ExaScaler-1.4 80Brick, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SC	50.32
2	5,331.79	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC/DL - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.1GHz, Infiniband FDR, NVIDIA Tesla K80	51.13
3	5,271.81	GSI Helmholtz Center	ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150	57.15
4	4,778.46	Institute of Modern Physics (IMP), Chinese Academy of Sciences	Sugon Cluster W780I, Xeon E5-2640v3 8C 2.6GHz, Infiniband QDR, NVIDIA Tesla K80	65.00
5	4,112.11	Stanford Research Computing Center	XStream - Cray CS-Storm, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR, Nvidia K80	190.00
6	3,856.90	IT Company	Inspur TS10000 HPC Server, Xeon E5-2620v3 6C 2.4GHz, 10G Ethernet, NVIDIA Tesla K40	58.00
7	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00
8	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00
9	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00
10	3,775.45	Internet Service	Inspur TS10000 HPC Server, Intel Xeon E5-2620v2 6C 2.1GHz, 10G Ethernet, NVIDIA Tesla K40	110.00

Aufbau eines Großrechners

- HPC System

Aufbau eines Großrechners

- HPC System
- Rack

Aufbau eines Großrechners

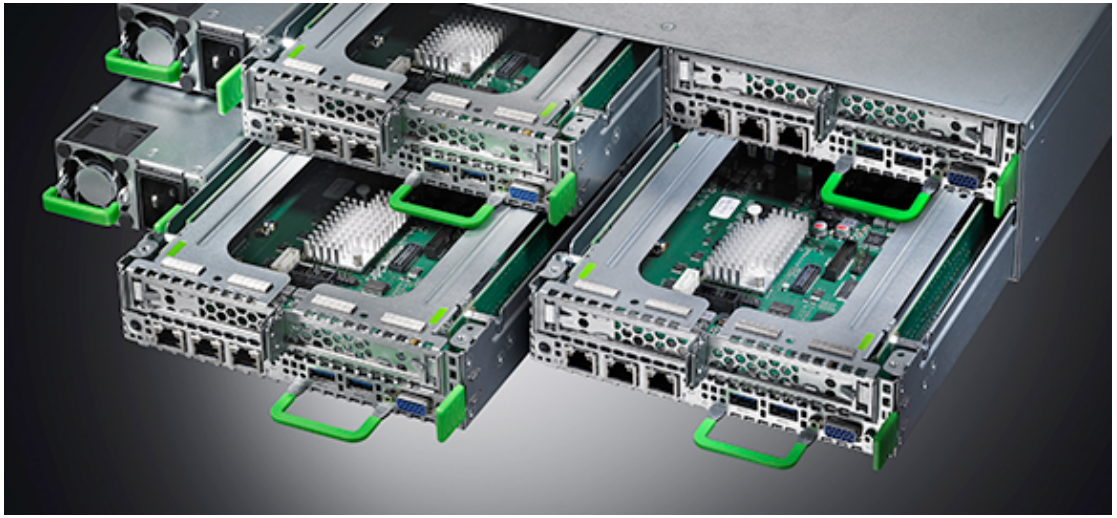
- HPC System
- Rack

Aufbau eines Großrechners

- HPC System
- Rack
- Blade Chassis

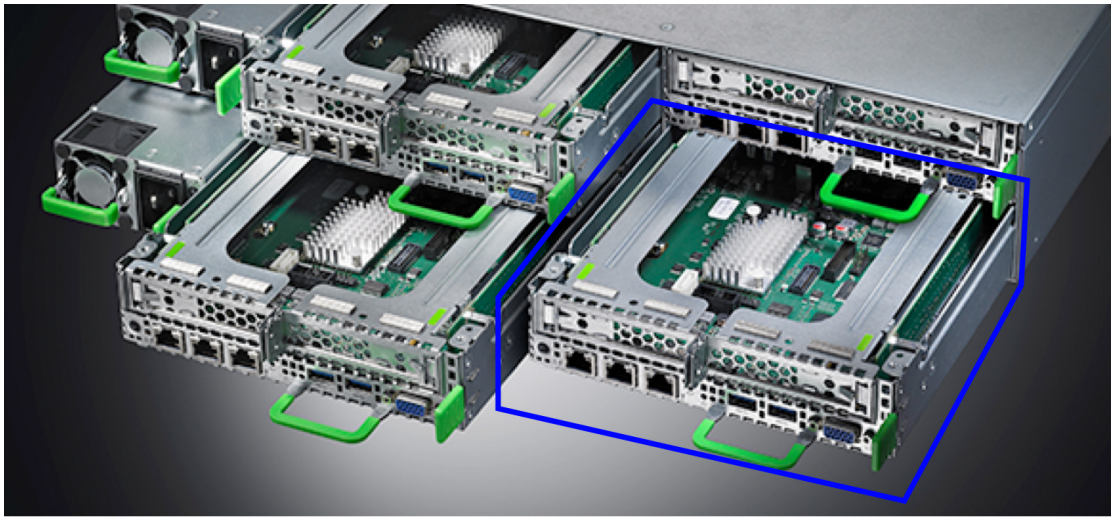
Aufbau eines Großrechners

- HPC System
- Rack
- Blade Chassis



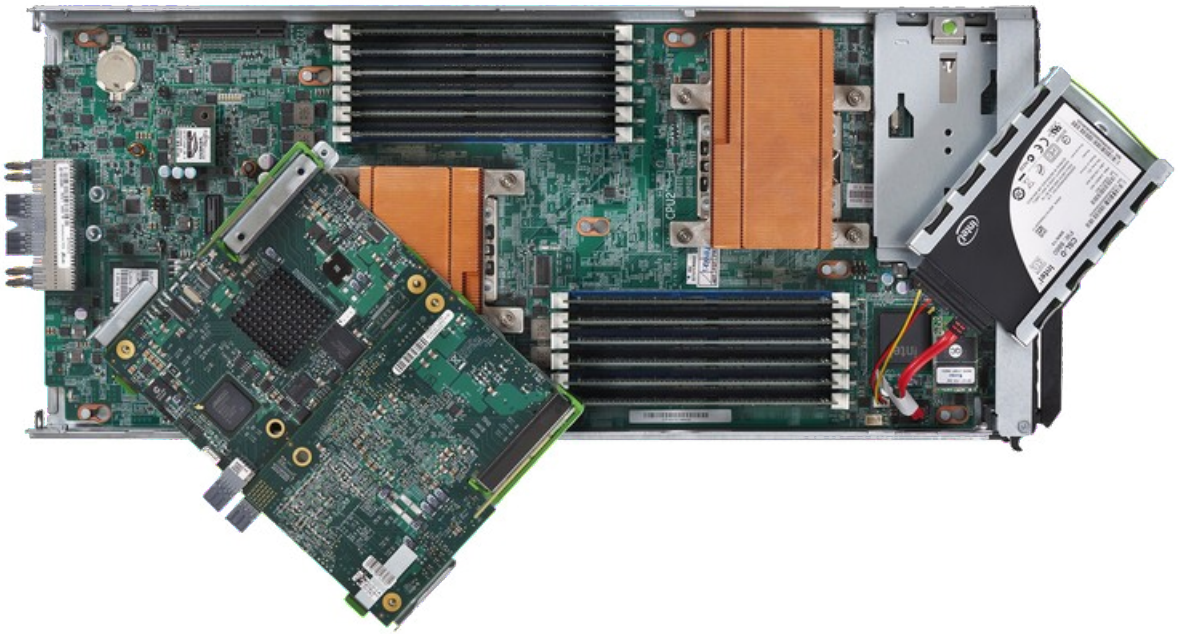
Aufbau eines Großrechners

- HPC System
- Rack
- Blade Chassis
- Blade



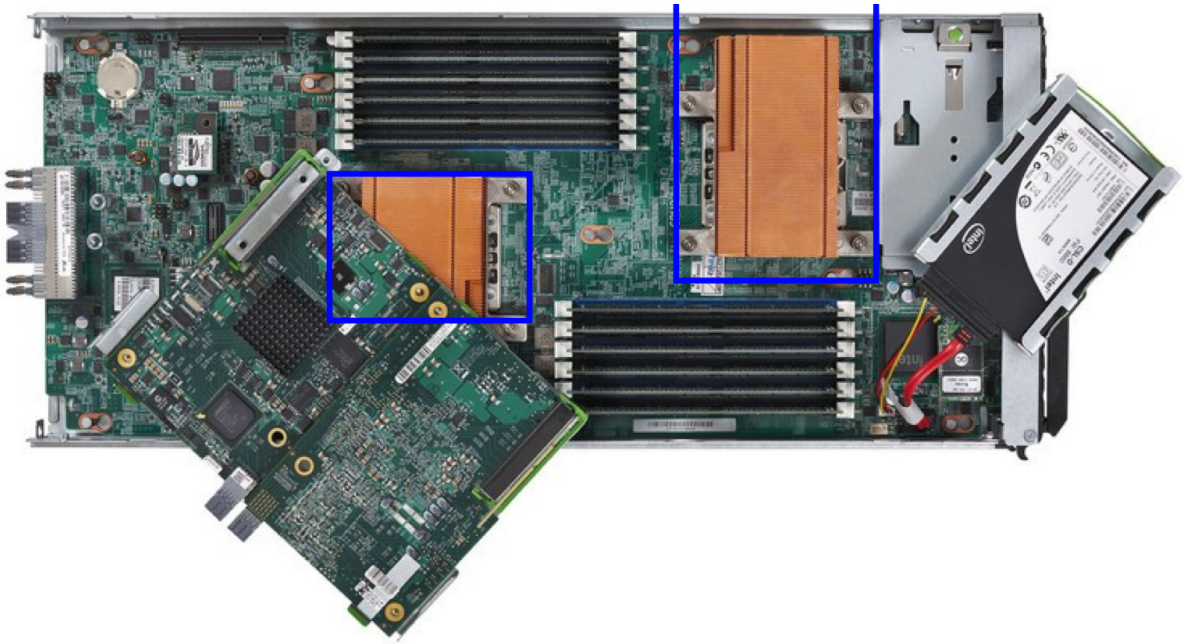
Aufbau eines Großrechners

- HPC System
- Rack
- Blade Chassis
- Blade



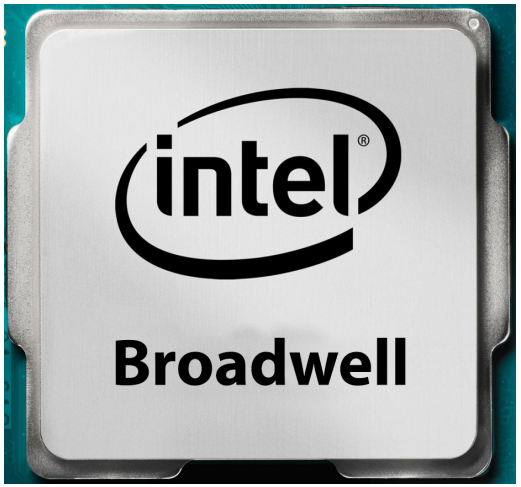
Aufbau eines Großrechners

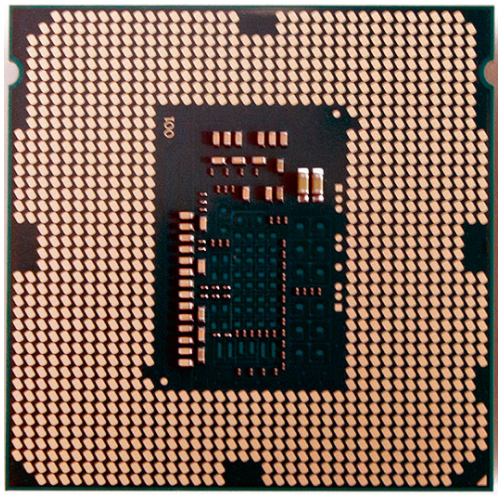
- HPC System
- Rack
- Blade Chassis
- Blade
- Prozessorchip



Aufbau eines Großrechners

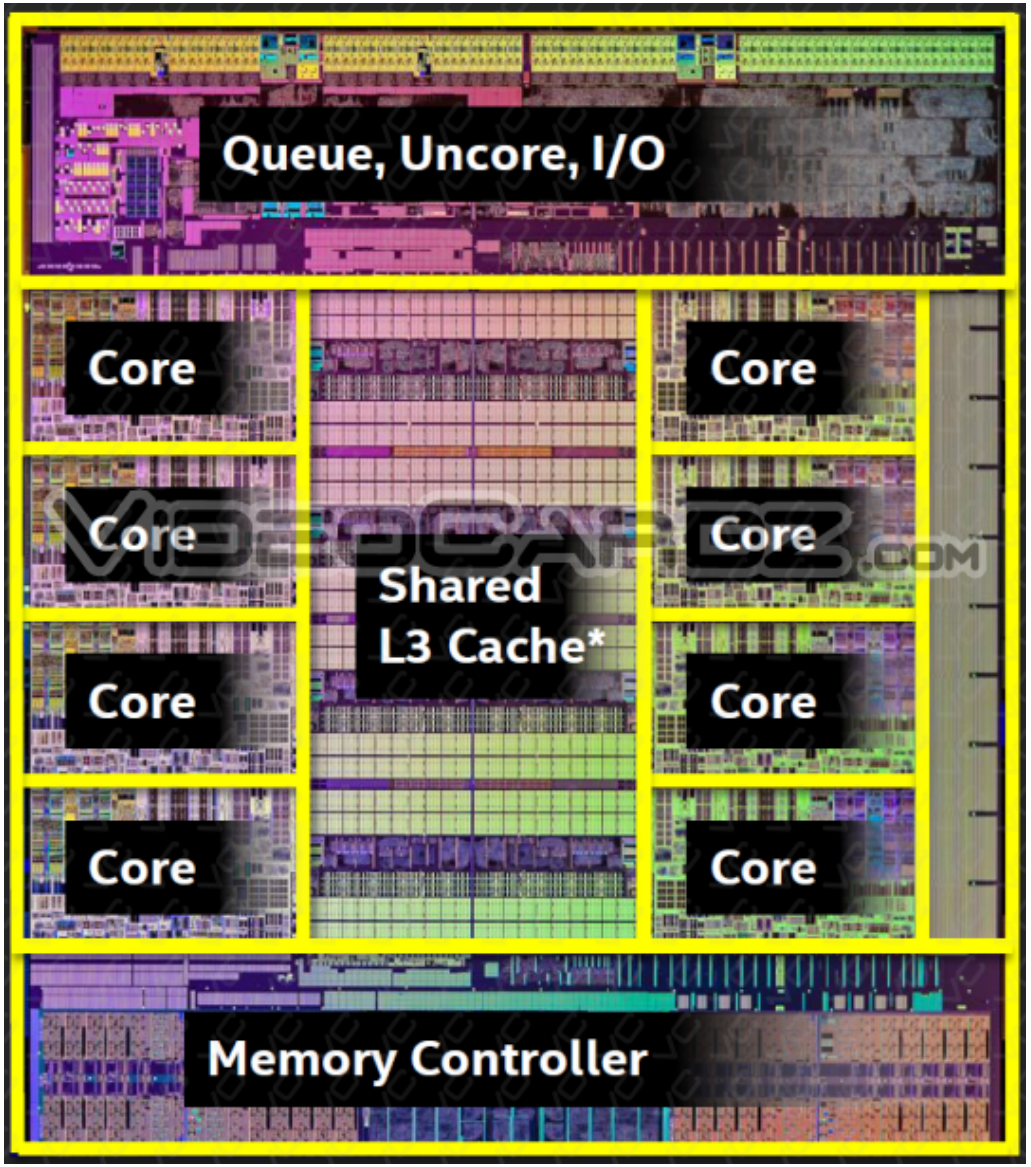
- HPC System
- Rack
- Blade Chassis
- Blade
- Prozessorchip



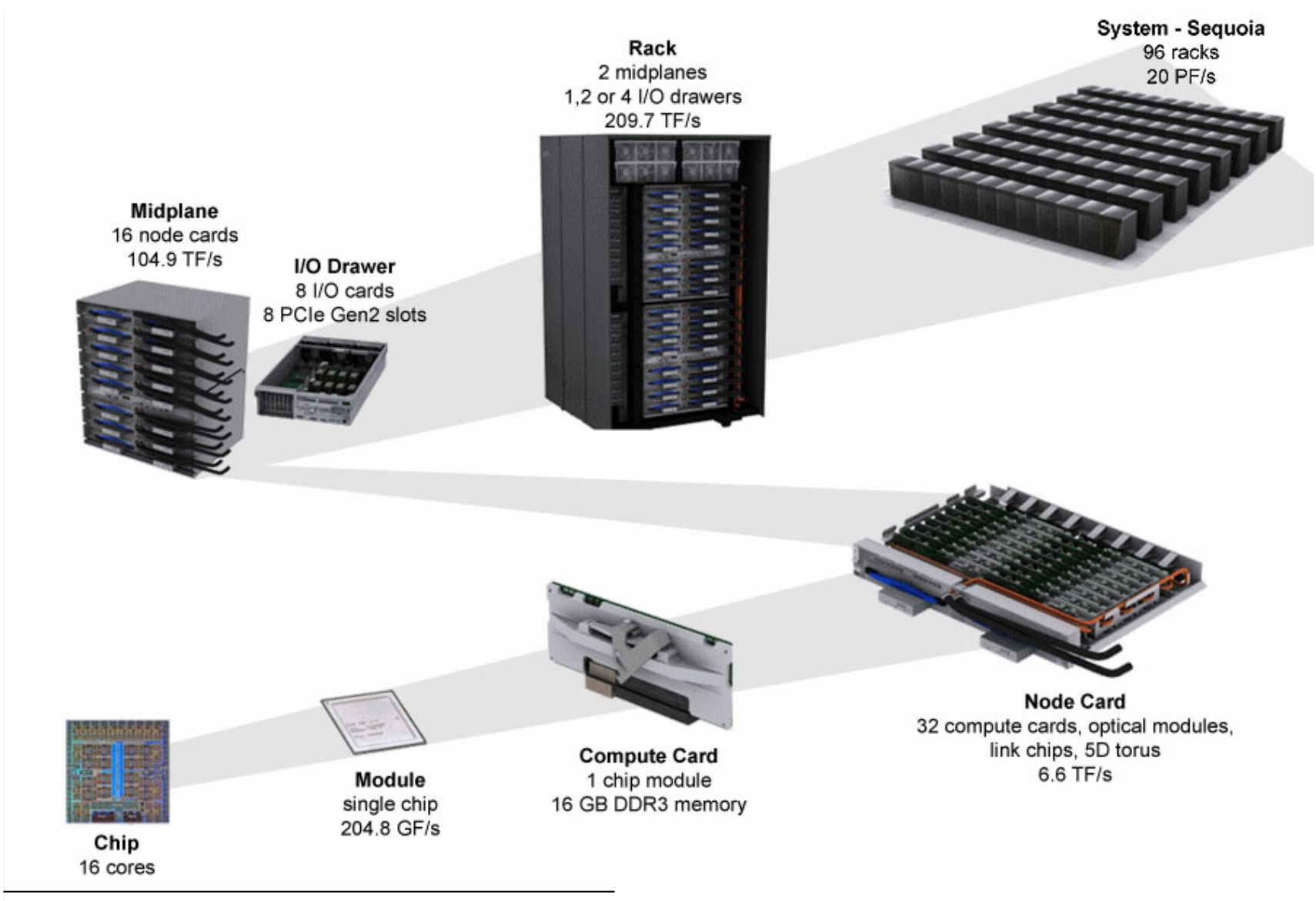


Aufbau eines Großrechners

- HPC System
- Rack
- Blade Chassis
- Blade
- Prozessorchip
- Cores



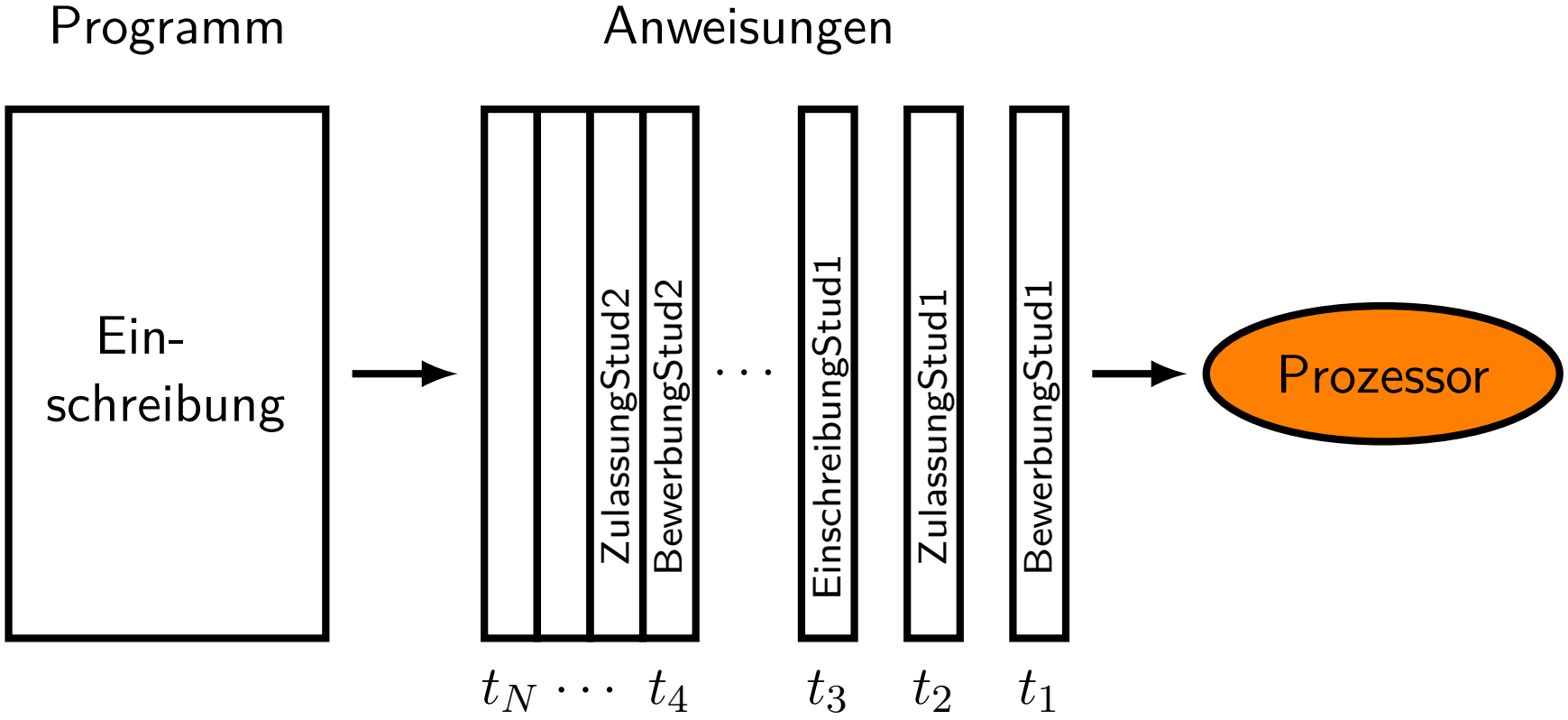
Aufbau eines Großrechners – IBM BG/Q System



LLNL-WEB-608955

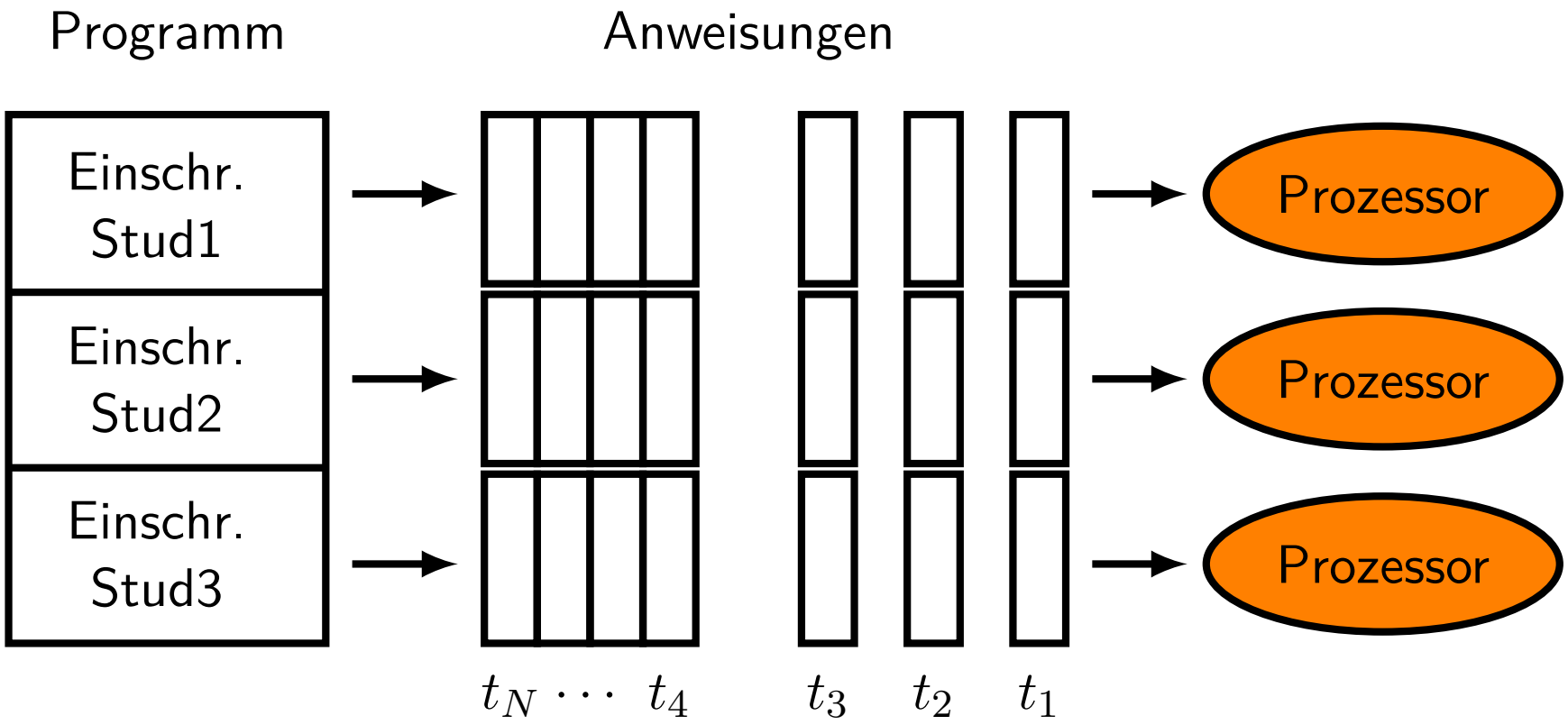
Wie verwende ich einen Supercomputer?

Serielles Rechnen – Beispiel: Einschreibung



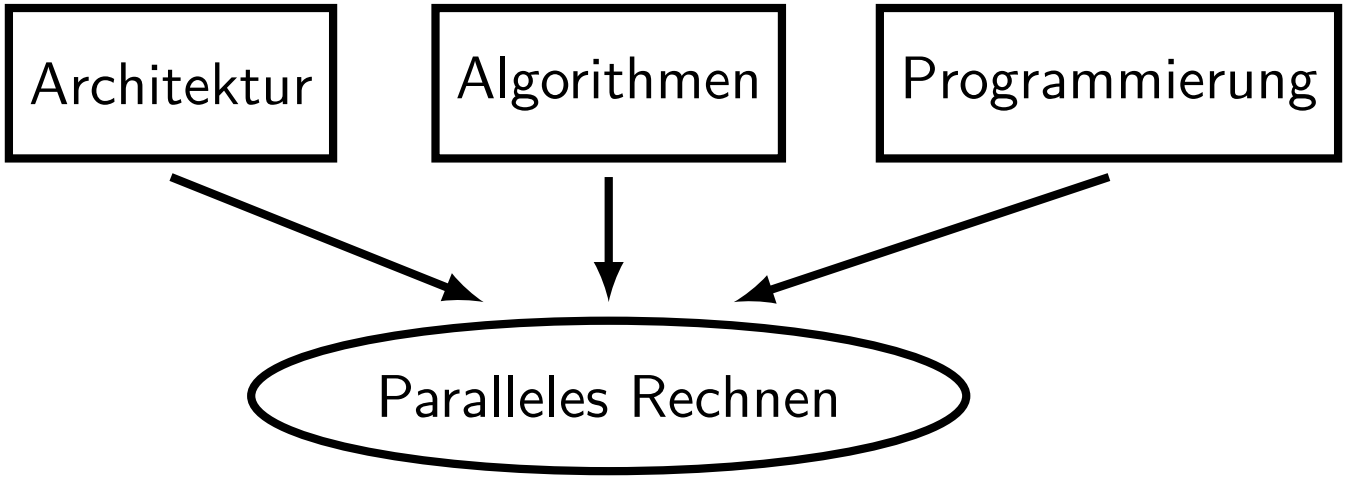
- Problem wird in eine diskrete Anzahl von Anweisungen geteilt
- Anweisungen werden sequentiell nacheinander von **einem Prozessor** abgearbeitet

Paralleles Rechnen – Beispiel: Einschreibung

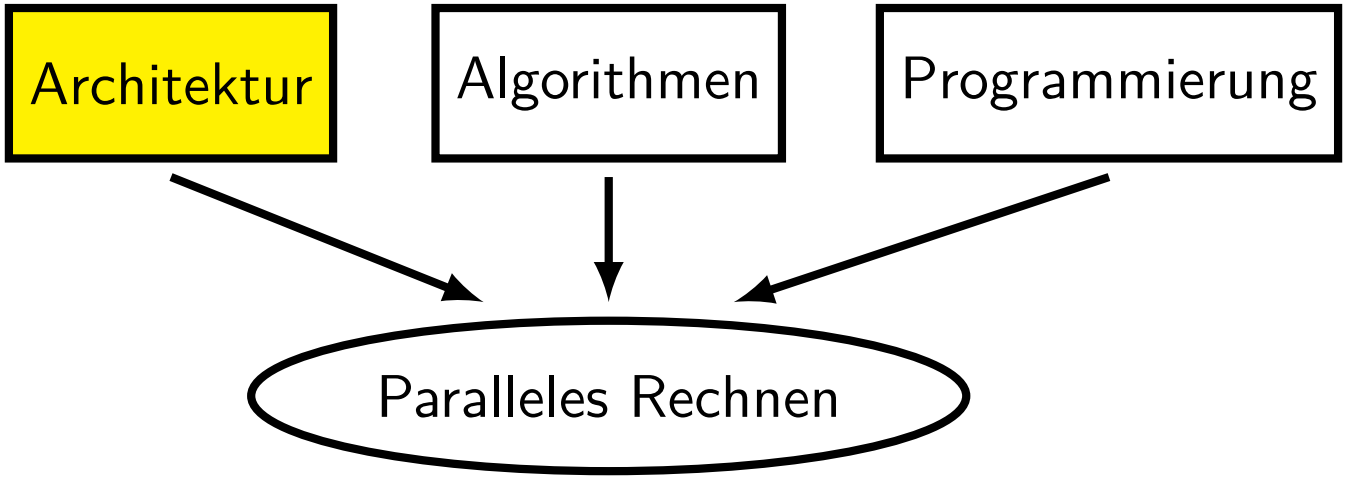


- Problem wird in eine diskrete Anzahl von Teilen geteilt
- Anweisungen von Problemteilen werden gleichzeitig auf **verschiedenen Prozessoren** abgearbeitet

Was bedeutet Parallelisierung?



Was bedeutet Parallelisierung?



Für die Aufteilung des Programms sind zwei Hardware-Ressourcen entscheidend

- Prozessoren
- Speicher

Parallelisierung bedeutet

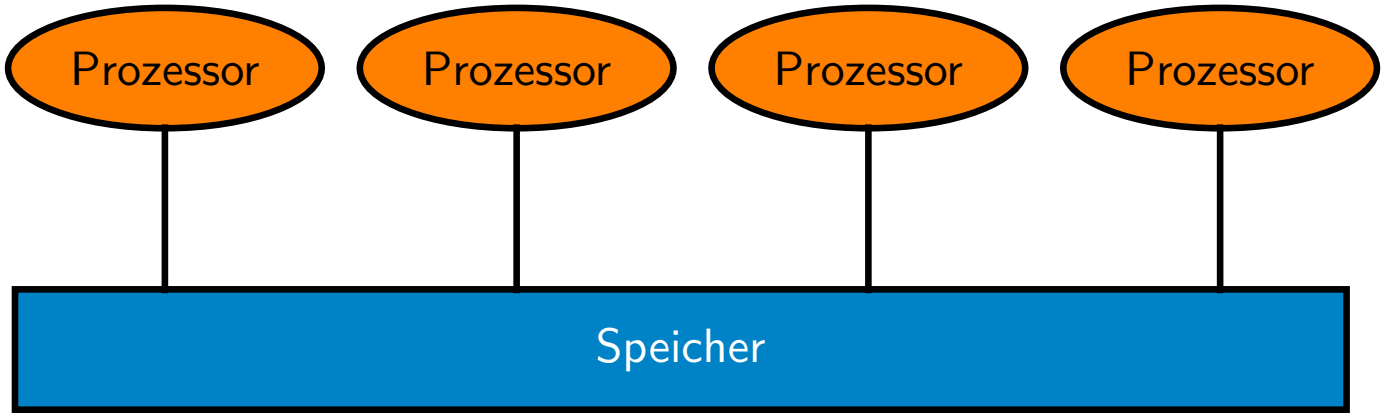
- Verteilen der Arbeit (Anweisungen) auf Prozessoren
- ggf. Verteilen der Daten

Shared memory – Multiprozessorsysteme

Alle Prozessoren greifen auf einen gemeinsamen Speicher zu

⊕ Änderungen im gemeinsamen Speicher sind für alle sichtbar

⊖ Speicher-Zugriffskonflike



SMP (Symmetrisches Multiprozessorsystem)

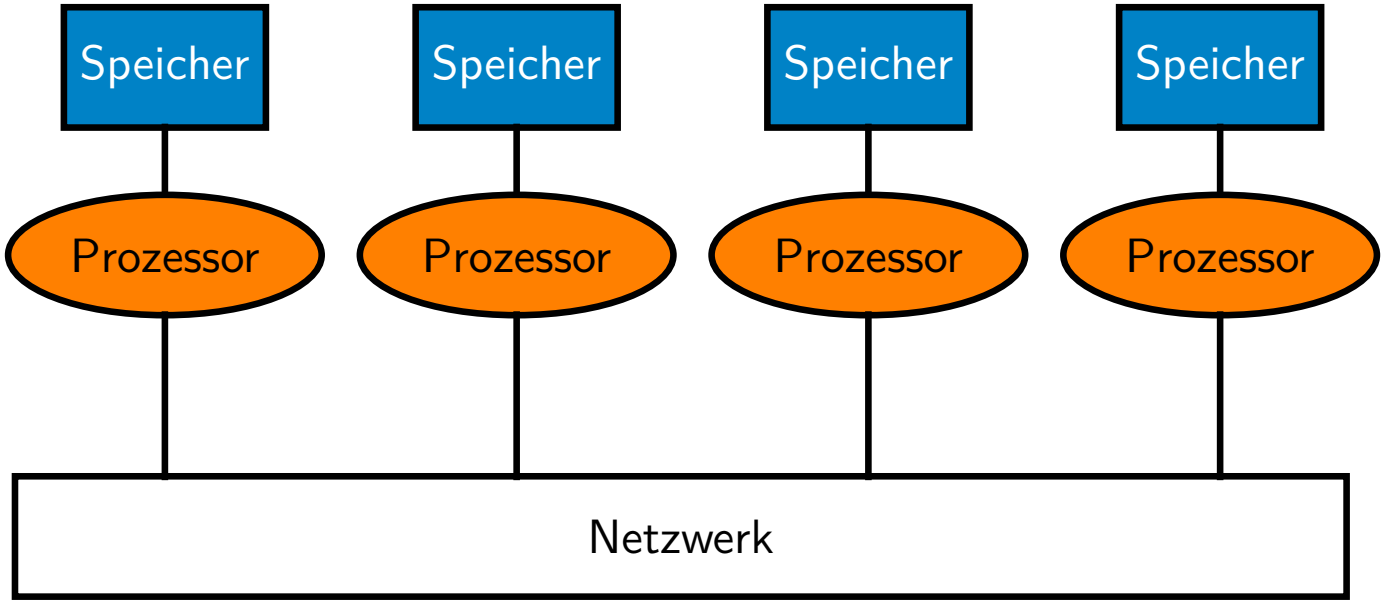
Shared memory Rechnerarchitektur, bei der mehrere Prozessoren einen gemeinsamen Adressraum teilen und gleichen Zugriff auf alle Ressourcen haben.

Distributed memory – Multirechnersysteme

Jeder Prozessor besitzt einen eigenen internen Speicher

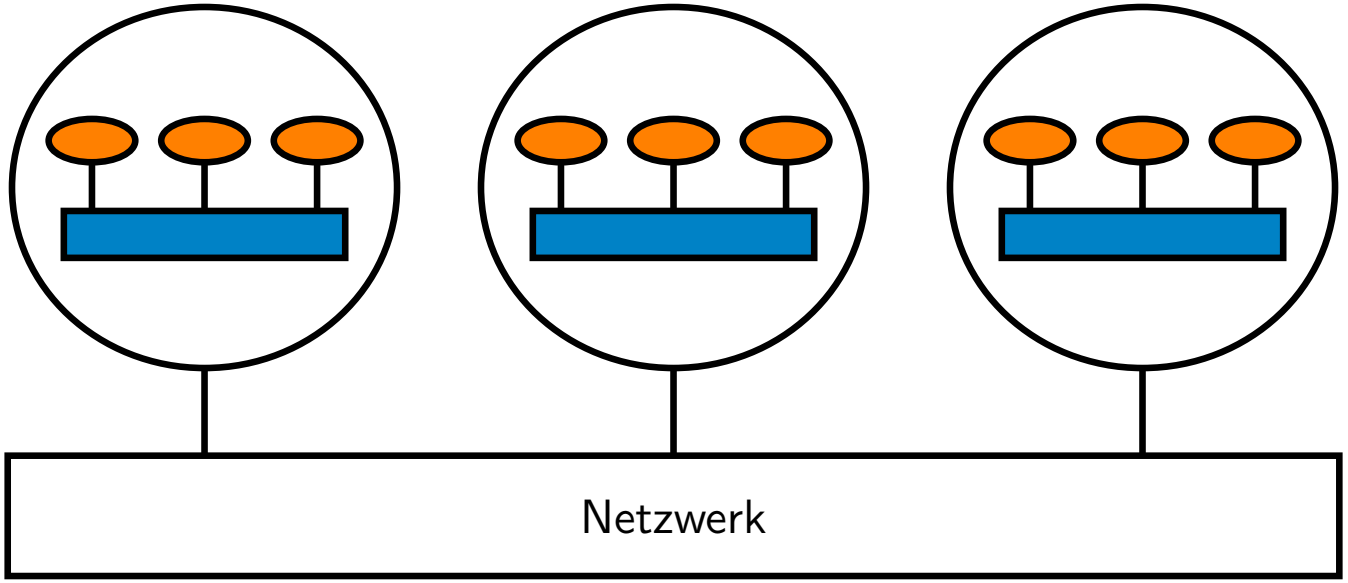
⊕ Keine Speicher-Zugriffskonflikte

⊖ Erfordert Datenaustausch über ein Netzwerk



Hybrid distributed-shared memory – moderne Architekturen

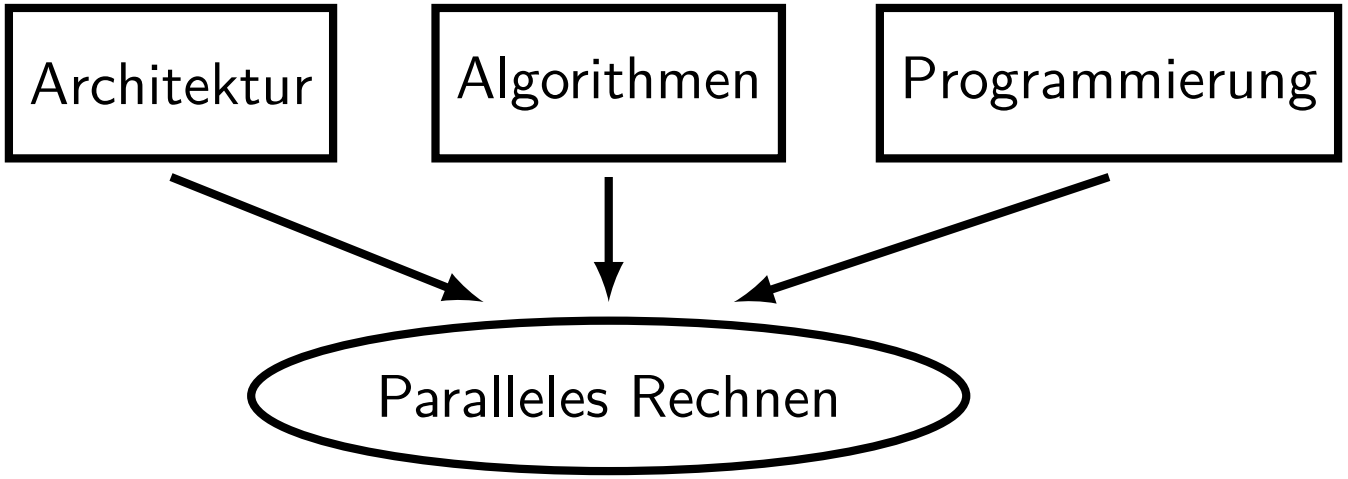
- Prozessoren mit gemeinsamen Speicher bilden einen Knoten
- Mehrere Knoten sind über ein Netzwerk verbunden



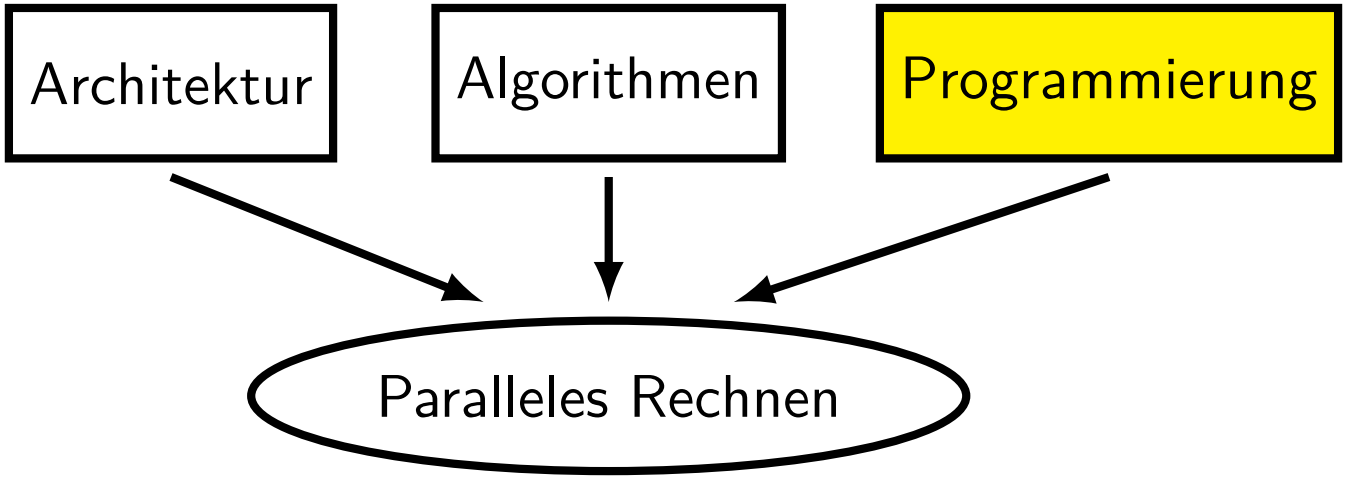
Node = Knoten

Verarbeitungseinheit in einem Rechenverbund bestehend aus Prozessoren, Speicher, Netzwerk-Interface, etc.

Was bedeutet Parallelisierung (II)?



Was bedeutet Parallelisierung (II)?



Warum funktioniert nicht einfach

```
gcc -Parallel mycode.c
```

???

Bis in die 1990er Jahre gab es keine Standards zur parallelen Programmierung $\rightarrow$ schwere und mühsame Programmierung, Plattform-abhängige parallele Codes

Programm vs. Prozess

Programm [Haase, 1999]

(statisch) exakte Niederschrift eines Algorithmus

Prozess [Haase, 1999]

(dynamisch) Folge von Aktionen

Shared memory – OpenMP

OpenMP standardisiert Shared-Memory-Programmierung

- Verteilung der Arbeit
- keine Datenaufteilung
- Benutzer definiert OpenMP-Direktiven
- Thread-basiert: Compiler erzeugt Threads
- Hauptsächlich Parallelisierung von Schleifen
- Standards
  - → Version 1.0 (Fortran: 1997, C/C++: 1998)
  - → Version 2.0 (Fortran: 2000, C/C++: 2002), 2.5 (2005)
  - → Version 3.0 (2008), 3.1 (2011)
  - → Version 4.0 (seit 2013)

Thread

Ausführungsstrang (Folge von Anweisungen) in der Abarbeitung eines Programms (`https://de.wikipedia.org/wiki/Thread_(Informatik)`)

Distributed memory – Message Passing Interface (MPI)

MPI standardisiert Distributed-Memory-Programmierung

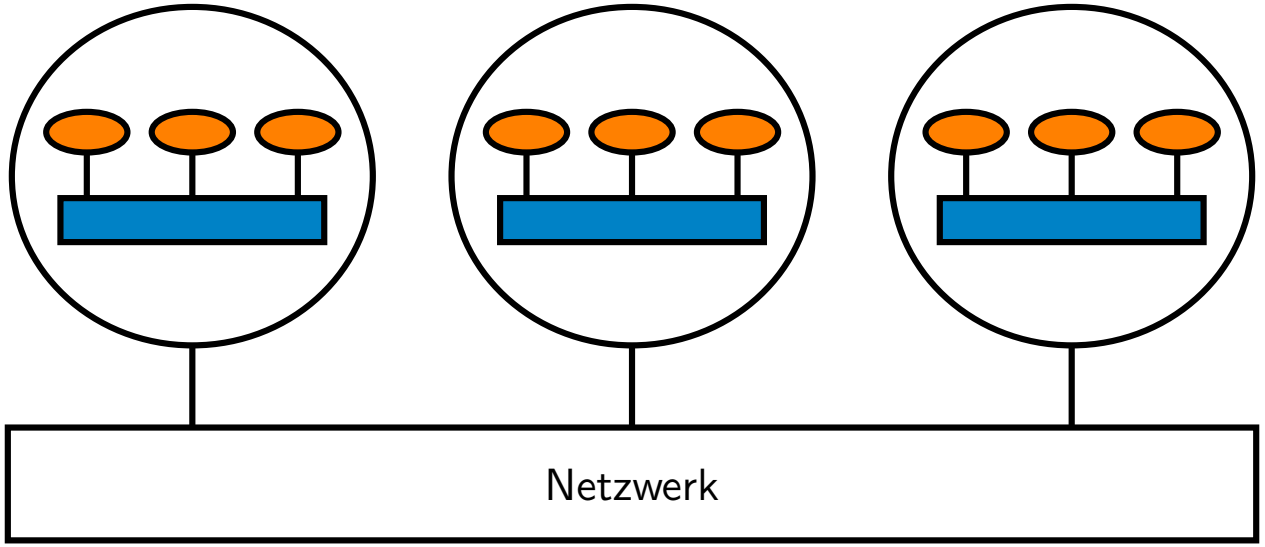
- Verteilung von Arbeit + Daten
- Datenaufteilung erfordert Kommunikation über spezielle send & receive Routinen (message passing)
- Benutzer ruft MPI-Bibliotheks-Routinen auf
- Prozess-basiert: Compiler generiert sequentiellen Code, jeder Prozessor führt ein (das gleiche) Sub-Programm aus
- Standards

  MPI-1 Version 1.0 (1994), 1.1 (1995), 1.2 (1997)
  MPI-2 Version 2.0 (1997), 2.1 (2008), 2.2 (2009)
  MPI-3 seit 2012

Message passing = Nachrichtenaustausch

Kommunikationsform, die zur Interprozesskommunikation eingesetzt wird und auf dem Senden und Versenden von Nachrichten basiert. ((https://de.wikipedia.org/wiki/Nachrichtenaustausch)

Programmiermodelle auf hybriden Systemen



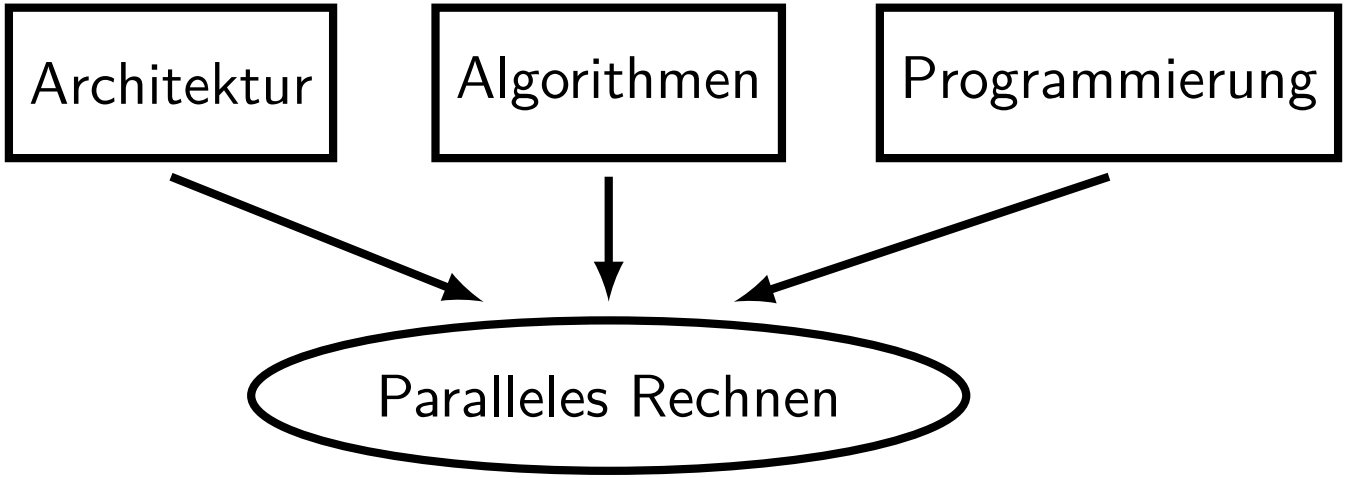
MPI

- jede CPU = ein MPI-Prozess

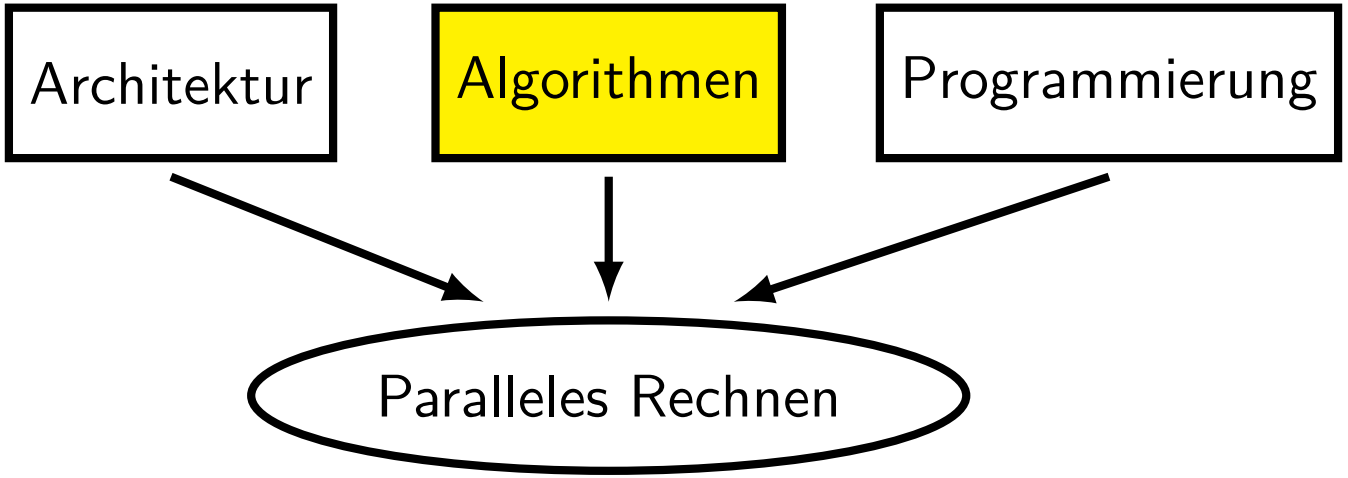
MPI + OpenMP

- jeder SMP-Knoten = ein MPI-Prozess
- OpenMP innerhalb eines SMP-Knotens

Was bedeutet Parallelisierung (III)?



Was bedeutet Parallelisierung (III)?



Bei der Umsetzung von Algorithmen spielen zwei Aspekte eine wichtige Rolle

1. Optimierung des seriellen Codes
   $\rightarrow$ beste Performance pro Prozess
2. Parallelisierung des optimierten seriellen Codes
   $\rightarrow$ effiziente Nutzung von Hardware-Ressourcen

Zusammenfassung

- Immer aufwendigere Simulation verlangen nach immer mehr Rechenleistung.
- Einzelne Prozessorkerne werden kaum noch leistungsstärker.
- Stattdessen stehen immer mehr parallel nutzbare Prozessoren in Parallelrechnern / Supercomputern zur Verfügung.
- Parallelrechner nutzen verschiedene Speichermodelle.
- Die Konsequenz: Algorithmen und Software müssen für Supercomputer entwickelt werden.
- Ein gutes, paralleles Programm ist effizient auf einem Core und nutzt viele Cores sinnvoll.

Ziel am Ende der Vorlesung:
Parallele und effiziente Software schreiben zu können

Literatur

- Gundolf Haase.
  Parallelisierung numerischer Algorithmen für partielle Differentialgleichungen.
  Teubner, 1999.
- Georg Hager, Gerhard Wellein.
  Introduction to High Performance Computing for Scientists and Engineers.
  CRC Press, 2011.
- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar
  Introduction to Parallel Computing, 2nd Edition.
  Addison Wesley, 2003.