



Cluster-Computing

Wie kann man viele vernetzte Standard-PC's
gemeinsam ein Problem lösen lassen?

Schülertag 2003

Prof. Dr. Peter Bastian

Prof. Dr. Thomas Ludwig



Wozu Superrechner?

Numerische Simulationen

Sogenannte „grand challenge applications“

- Modellierung natürlicher/technischer Systeme mittels mathematischer Gleichungen
- Numerische Lösung dieser Gleichungen auf Superrechnern

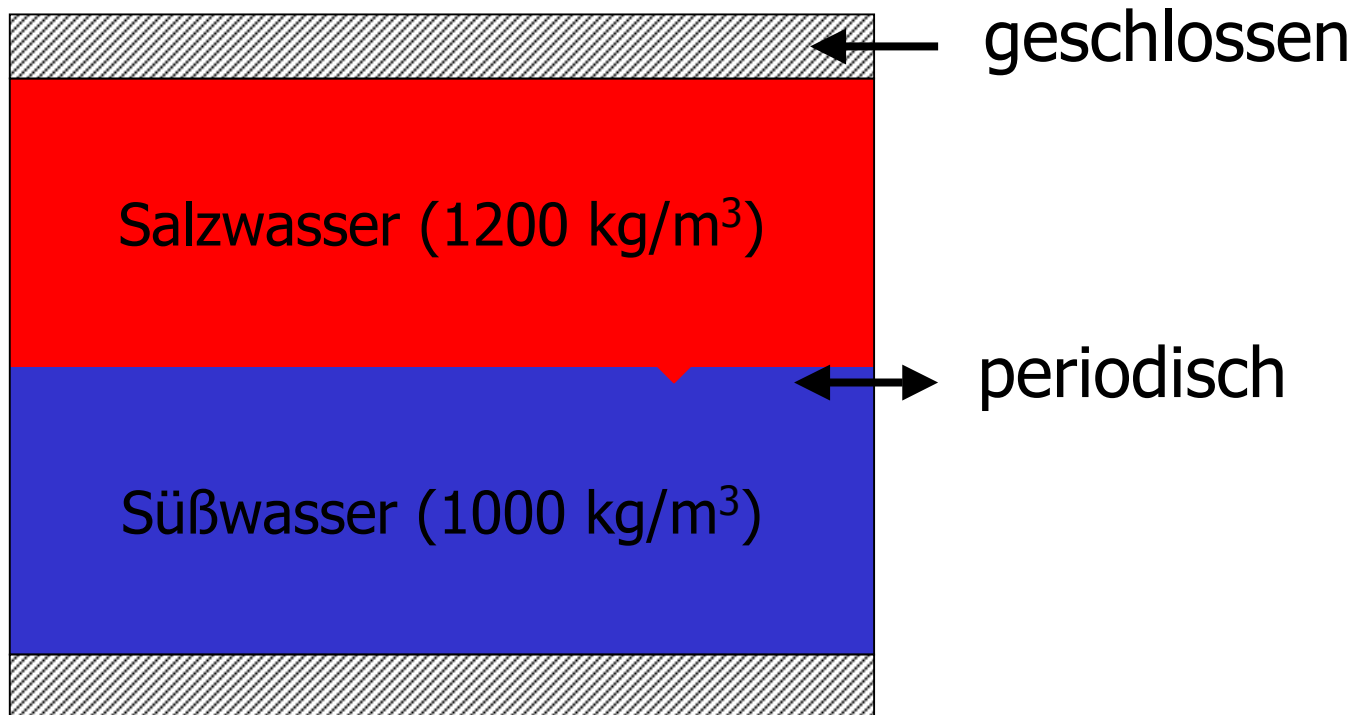
Beispiele

- Proteinfaltung
- Wettervorhersage
- Astrophysik
- ...

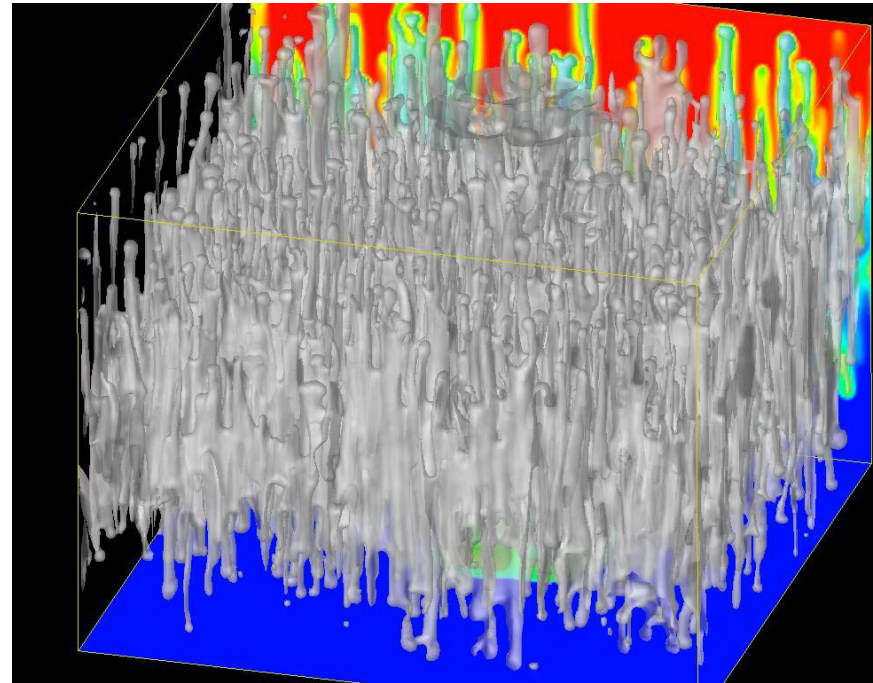
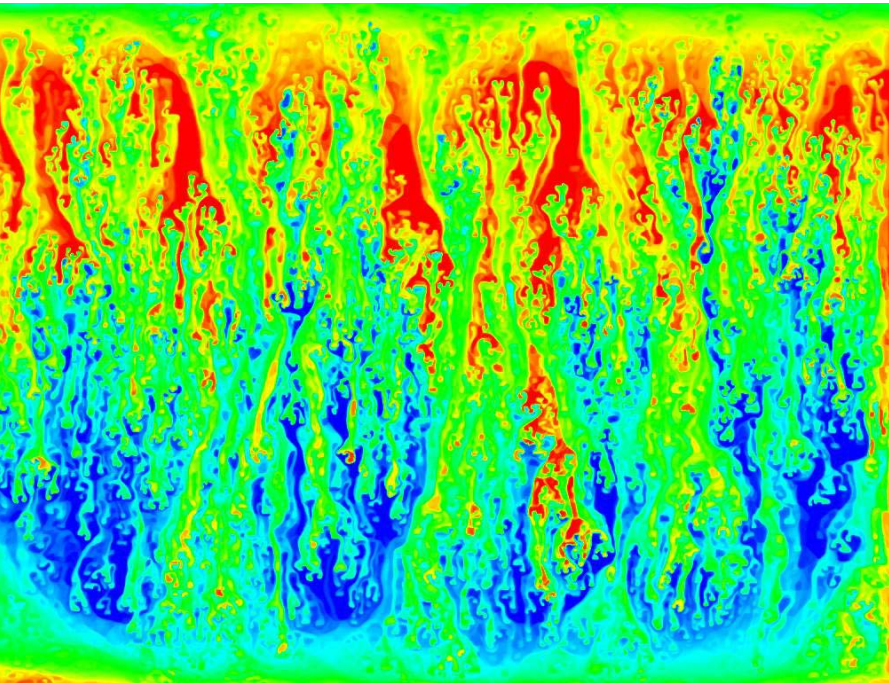
Beispielproblem

Dichtegetriebene Grundwasserströmung

Ähnlich zu Ozean, Erdmantel, Atmosphäre, Kaffeetasse ...



Filme 2D/3D



Berechnung der Strömung

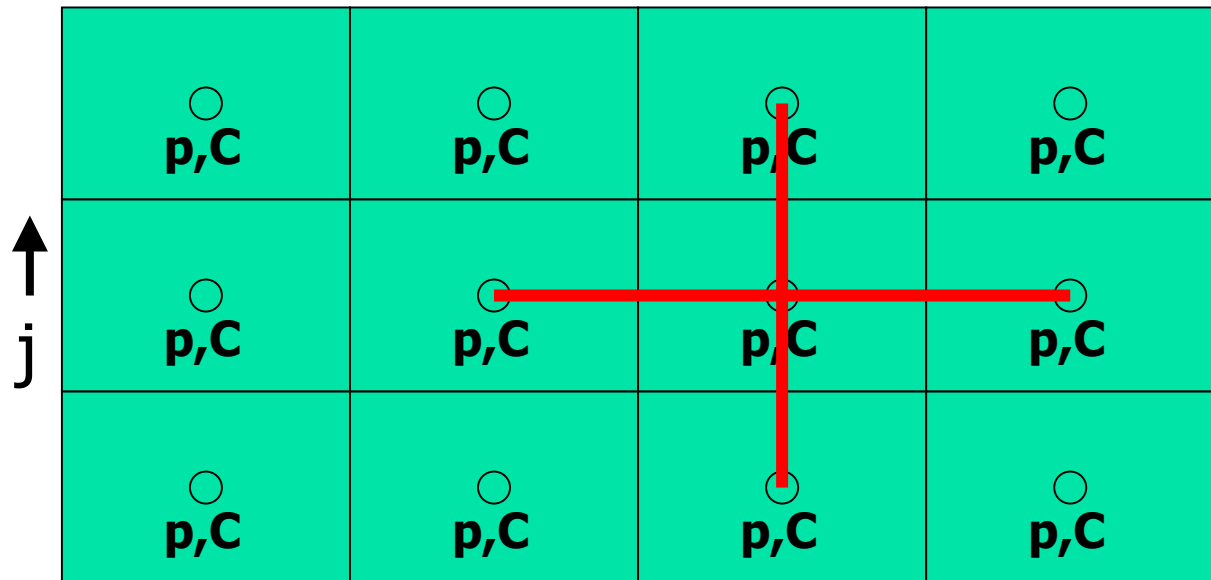
Lösen großer linearer Gleichungssysteme

P: Druck, C: Salzkonzentration

Vernüpfung der Zellenwerte untereinander:

$$4p_{i,j} = p_{i-1,j} + p_{i+1,j} + p_{i,j-1} + p_{i,j+1}$$

i →



Anzahl Rechenoperationen

3D Beispiel: $N=256 \times 256 \times 192 = 12.582.912$ Zellen

Gleichungssystem lösen: $\approx N \times 3000$ Gleitkommaoperationen (Flop) für „genaue“ Lösung

$\Rightarrow \approx 4 \times 10^{10}$ arithmetische Operationen pro Schritt

4000 Zeitschritte

$\Rightarrow \approx 1.5 \times 10^{14}$ arith. Operationen für die Simulation

Mit PentiumIV mit 2.4GHz $\approx 3.5 \times 10^8$ Flop/s

$\Rightarrow \approx 119$ Stunden Rechenzeit

Speicherbedarf: $N \times 36$ Byte ≈ 500 MByte

Realistische Simulation mit Auflösung wie 2D:

$1024 \times 1024 \times 768 \times 36$ Byte ≈ 28 GByte

Superrechner – gestern und heute

CDC 6600, Control Data, 1964, 7M\$

10MHz Takt, \approx 512KByte Speicher, \approx 3MFlop/s

Cray 1A, Cray Research, 1976, 8M\$

80MHz Takt, 8MByte Hauptspeicher, \approx 20MFlop/s

Cray X-MP/48, Cray Research, 1986, 14.6M\$

118MHz, 4 Prozessoren, 64MByte Speicher,
55MFlop/s pro Prozessor

Cray C90, Cray Research, 1996

250MHz, 5GFlop/s, 16 Prozessoren, 2GByte
Hauptspeicher

Superrechner – gestern und heute...

ASCI Red, Intel, 1997

200MHz Pentium Pro, 9152 Prozessoren,
1.3TFlop/s

Pentium4, Intel, 2002, 1.500\$

2.4GHz, 4.8GFlop/s nominell, 1GByte Speicher

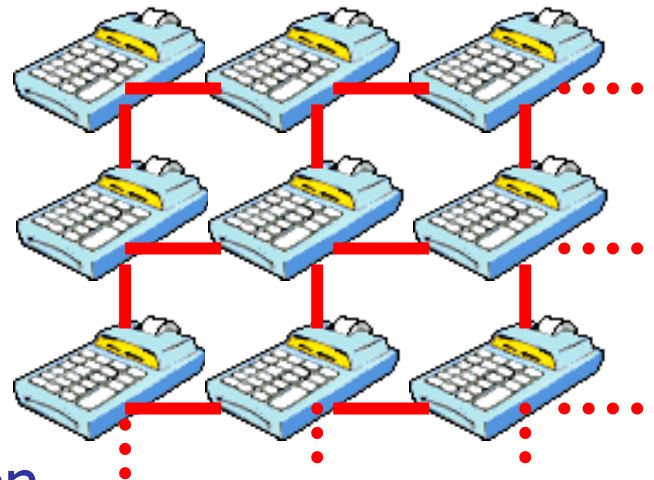
Heute Superrechner – morgen Arbeitsplatzrechner

- Heutige PC-Prozessoren nutzen Technologien, die über 30 Jahre in Superrechnern entwickelt und erprobt wurden
- Moore'sche Vermutung: Bei gleichen Kosten verdoppelt sich die Leistung von Mikroprozessoren alle 18 Monate
- Konsequenz: es steht uns immer mehr Leistung im Einzelprozessor zur Verfügung

Cluster-Computing

Problem: Rechenleistungsbedarf geht weit über das hinaus, was der Einzelprozessor liefern kann

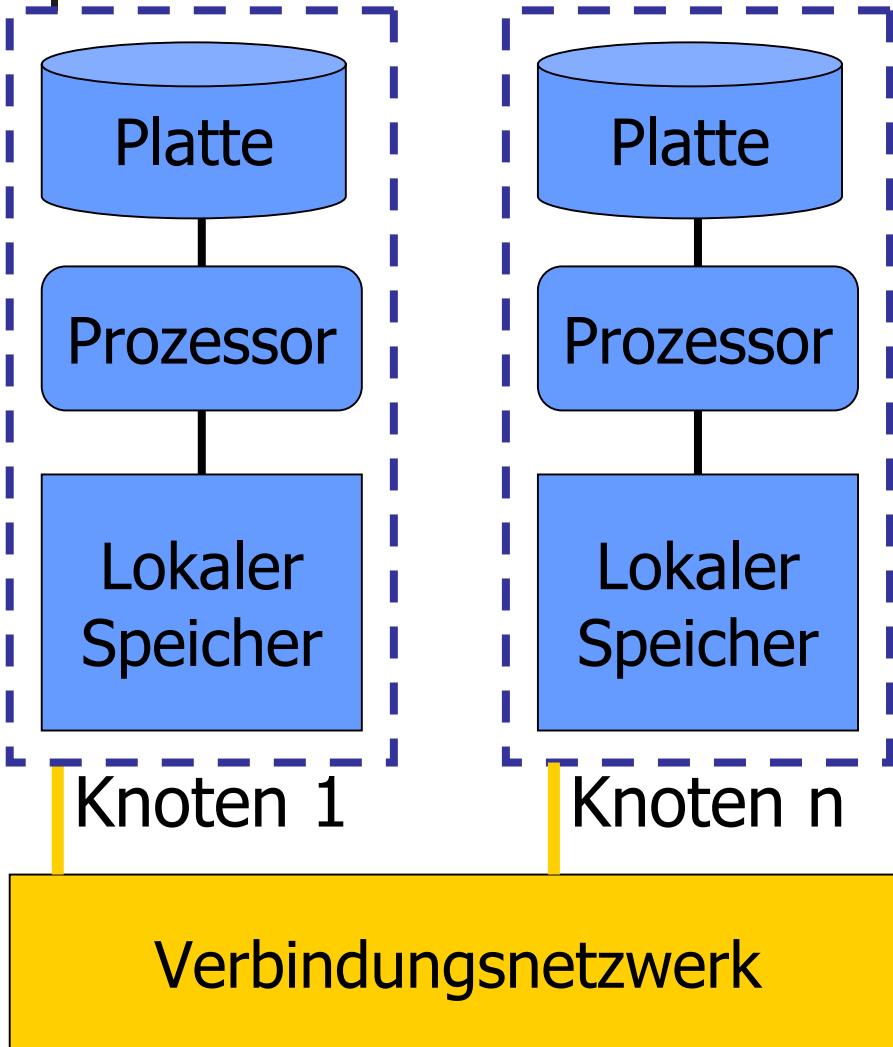
Idee: Nicht mehr nur ein Rechner, sondern viele vernetzte Rechner



Warum gerade so?

- Einzelrechner sind billig zu haben
- Ausbau gemäß den finanziellen Mitteln
- Einzelrechner stehen teils ungenutzt herum

Cluster-Computing...



Jetzt n Rechnerknoten

- Evtl. mehrere Prozessoren je Knoten
- Leistungsfähiges Verbindungsnetzwerk
- Hohe Gesamtleistung
- Hohe Ausfallrate von Einzelteilen
- Komplizierte Programmierung
- Komplizierter Betrieb



Das HELICS-Cluster

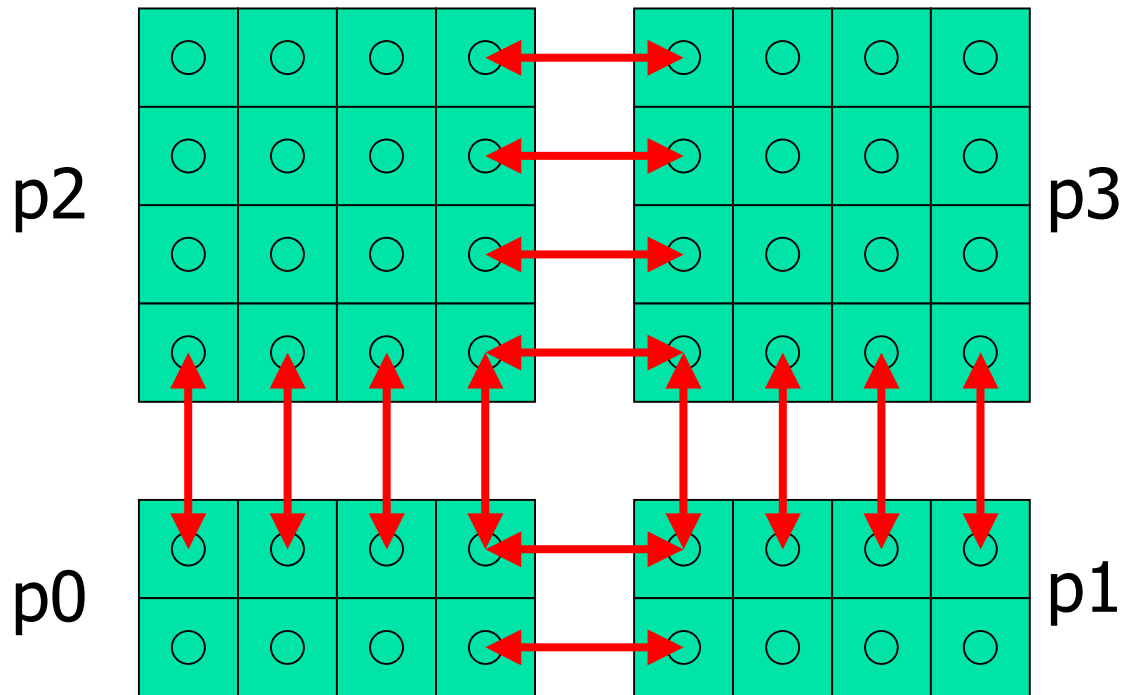
- Heidelberger Linux Cluster
- 256 Knoten à 2 AMD-Prozessoren à 1.4 GHz
- 2 GByte Hauptspeicher pro Knoten
- Mehrere TByte Hintergrundspeicher
- Myrinet Glasfaser-Vernetzung (2GBit)
- Linux-Betriebssystem
- 1.4 TFlop nominelle Gesamtleistung
0.83 TFlop gemessen; z.Zt. Rang 80

HELICS-Cluster (vordere Hälfte)



Parallele Programmierung

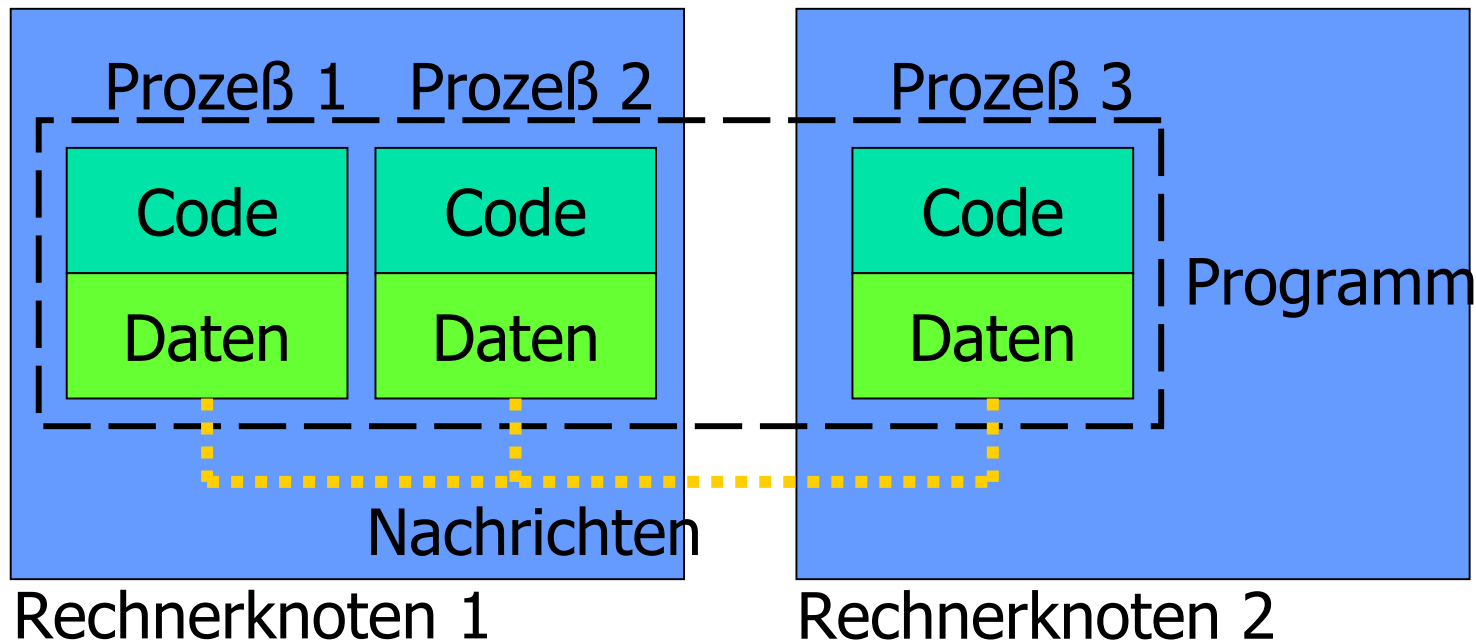
- Teile Zellen auf Prozessoren auf (Datenparallelität)
- Füge Kommunikation zwischen Prozessoren ein



Parallele Programmierung...

Programm verteilt sich über alle Knoten

- Identischer Programmcode in jedem Prozeß
- Unterschiedliche Daten in jedem Prozeß





Parallele Programmierung...

Probleme

- Zusatzlast durch Kommunikation
 - Verlust an verfügbarer Gesamtrechenleistung
- Lastungleichheit
 - Unter-/überlastete Knoten
- Fehlersuche
 - Neue Fehlertypen: zeitabhängige Fehler

Fehlerfreie und effiziente parallele
Programmierung erfordert viel Erfahrung



Optimierung und Grenzen

Frage: Wenn 1 Prozessor zur Lösung die Zeit t braucht, welche Zeit brauchen 10 Prozessoren?

Antwort: Mit etwas Glück $t/10$

Frage: Wie lange brauchen 512 Prozessoren?

Frage: Wie lange brauchen 10.000 Prozessoren?

Erkenntnis:

- Parallelisierung ist immer nur bis zu einer bestimmten Prozessoranzahl gewinnbringend
- Jenseits davon überwiegt der Koordinationsaufwand den Nutzen der Parallelisierung

Schnelle Rechner sind nicht alles – Effiziente Algorithmen

Mit p Rechnern maximal p -mal schneller

Mit vielen Prozessoren will man große Probleme lösen

Im Beispiel: Man möchte immer mehr Zellen N rechnen

N	Gauß ($2/3N^3$)	Mehrgitter ($100N$)
1.000	0.66s	10^{-4} s
10.000	660s	10^{-3} s
100.000	7.6 Tage	10^{-2} s
10^6	21 Jahre	10^{-1} s
10^7	21.000 Jahre	1s

Bei einem Rechner mit 1GFlop/s



Aktuelles und Zukünftiges

Giga-Flop/s: Heute schon im Einzelprozessor

Tera-Flop/s: Seit ca. 1998 erreicht

Peta-Flop/s: Für ca. 2008 hochgerechnet

Dann auch Tera-Flop/s im Notebook

Spitzenreiter heute:

NEC's Earth Simulator

41TFlop/s nominelle Leistung

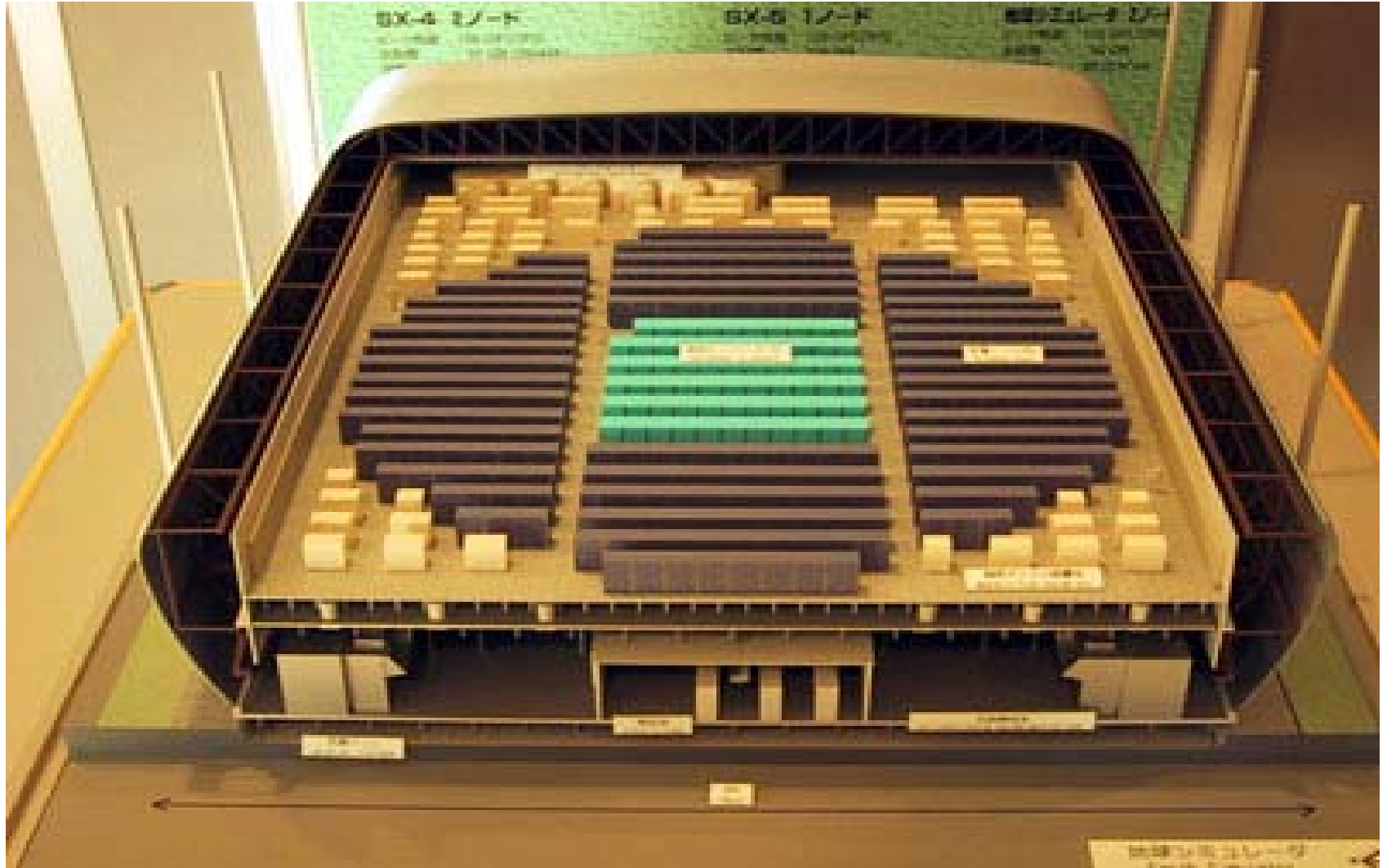
36TFlop/s erbrachte Leistung mit LINPACK



NEC's Earth Simulator

- 640 Knoten
- Zu je 8 Vektorprozess.
- 5120 Prozessoren
- 0.15mikron Kupfer
- 400M\$
- Zum Zwecke der Klimaforschung etc.
- 10TByte Hauptspeicher
- 700TByte Platten
- 1.6PByte Bandspeicher
- 83.000 Kupferkabel
- 2.800km Kabel
- 3250qm
- Eigenes Kraftwerk
- Erdbebensicher

Earth Simulator Gebäudeaufbau





Aktuelles und Zukünftiges...

Kleines Gedankenspiel

1 Peta-Flop/s erreichbar durch 30 Earth Simulator

Sozusagen: Earth-Simulator-Cluster

- Einzelgebäude mit 100.000qm ??
- 30 Kraftwerke ??
- 12 Milliarden US\$??

Konsequenz

Keine PetaFlop/s-Leistung mit heutiger Technologie

Grenze der Cluster-Technik durch Kosten- und Energieaufwendungen gegeben



Weitere Informationen

Die Top-500-Liste der stärksten Rechner der Welt

www.top500.org

HELICS-Parallelrechner der Universität Heidelberg

helics.uni-hd.de

Bildergalerie alter Superrechner

www.scd.ucar.edu/computers/gallery

Grand Challenges – Die großen Herausforderungen

www.nhse.org/grand_challenge.html

Links zu den Referenten

Peter Bastian: hal.iwr.uni-heidelberg.de

Thomas Ludwig: pvs.iwr.uni-heidelberg.de