

Rechnerstrukturen

Vorlesung im Sommersemester 2009

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik



- **Kapitel 2: Parallelismus auf Befehlsebene**

2.4: Thread-Level Parallelismus, Multithreading

- Grundsätzliche Aufgabe beim Prozessorentwurf:
 - Reduzierung der Untätigkeits- oder Latenzzeiten
 - Entstehen bei Speicherzugriffen, insbesondere bei Cache-Fehlzugriffen
 - Bei speichergekoppelten Multiprozessoren, wenn auf nicht-lokalen Speicher zugegriffen wird
 - Synchronisation von parallelen Kontrollfäden (Threads)
 - Lösung: parallele Ausführung mehrerer Kontrollfäden

- **Mehrfädige Prozessortechnik**
 - Gegeben mehrere ausführbereite Kontrollfäden, Threads
 - Ziel: Parallele Ausführung mehrerer Kontrollfäden
 - Voraussetzung:
 - Mehrere Kontrollfäden sind geladen
 - Kontext muss für jeden Thread gesichert werden können
 - Mehrere getrennte Registersätze auf Prozessorchip
 - Mehrere Befehlszähler
 - Getrennte Seitentabellen
 - Threadwechsel, wenn gewartet werden muss

- **Mehrfädige Prozessortechnik**

- Cycle-by-cycle Interleaving (feingranulares Multithreading)

- Eine Anzahl von Kontrollfäden ist geladen.
- Der Prozessor wählt in jedem Takt einen der ausführungsbereiten Kontrollfäden aus.
- Der nächste Befehle in der Befehlsreihenfolge des ausgewählten Kontrollfadens wird zur Ausführung ausgewählt.
- **Beispiele**
 - Multiprozessorsysteme HEP, Tera
- **Nachteil:**
 - Die Verarbeitung eines Threads kann erheblich verlangsamt werden, wenn er ohne Wartezeiten ausgeführt werden kann

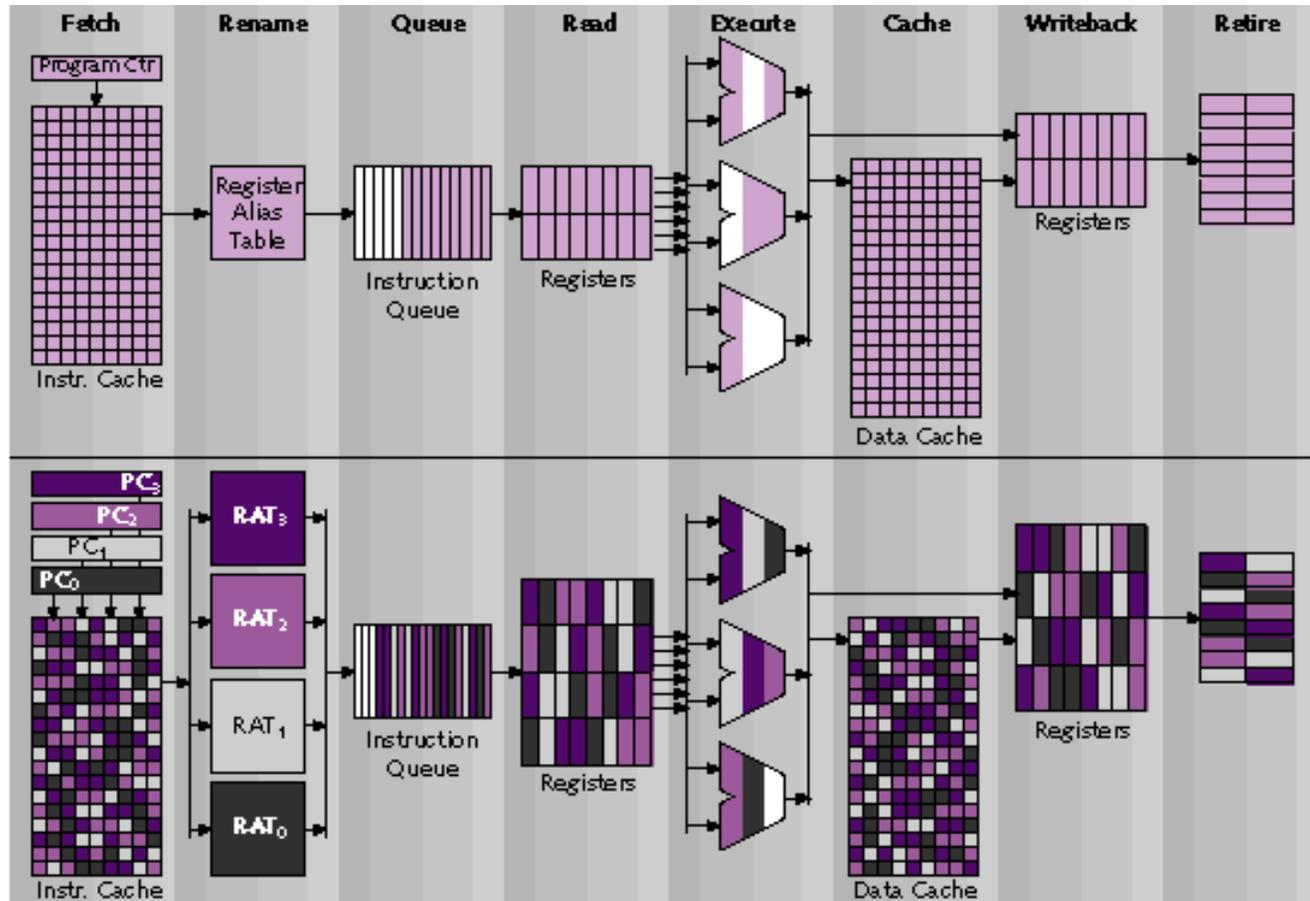
- Mehrfädige Prozesstechnik

- Block Interleaving

- Befehle eines Kontrollfadens werden so lange ausgeführt, bis eine Instruktion mit einer langen Latenzzeit ausgeführt wird. Dann wird zu einem anderen ausführbaren Kontrollfaden gewechselt.
- Vorteil:
 - Die Bearbeitung eines Threads wird nicht verlangsamt, da beim Warten ausführungsbereiter Thread gestartet wird
- Nachteil:
 - Bei Thread-Wechsel Leeren und Neustarten der Pipeline,
 - Nur bei langen Wartezeiten sinnvoll

- **Mehrfädige Prozesstechnik**
 - **Simultaneous Multithreading**
 - Mehrfach superskalärer Prozessor
 - Die Ausführungseinheiten werden über *eine* Zuordnungseinheit aus mehreren Befehlsuffern versorgt.
 - Jeder Befehlsuffter stellt einen anderen Befehlsstrom dar.
 - Jedem Befehlsstrom ist eigener Registersatz zugeordnet.

- Mehrfädige Prozesortechnik
 - Simultaneous Multithreading

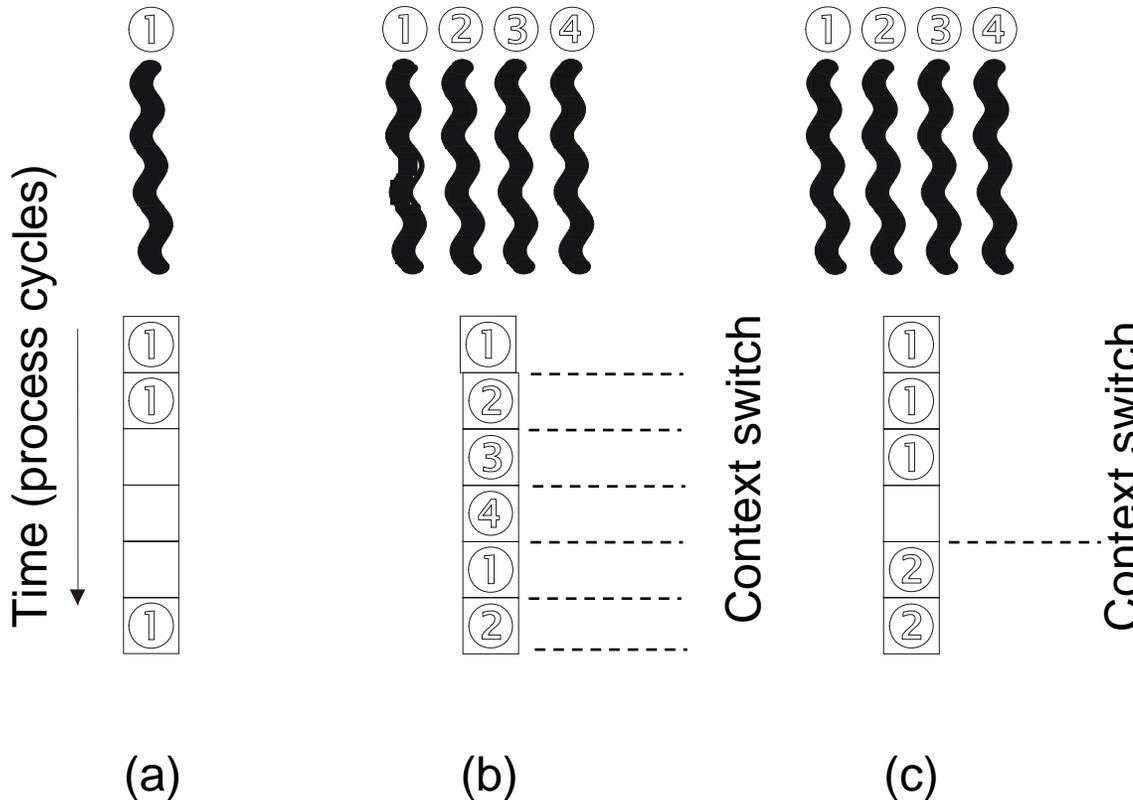


- **Mehrfädige Prozesstechnik**
 - **Simultaneous Multithreading: Diskussion**
 - Abwägen zwischen Geschwindigkeit eines Threads und dem Durchsatz vieler Threads
 - Ein bevorzugter Thread
 - » Allerdings kann dies auf Kosten des Durchsatzes gehen, da Befehle anderer Threads möglicherweise nicht bereit stehen
 - Mischen vieler Threads:
 - » Geht möglicherweise zu Lasten der Leistung der einzelnen Threads

- **Mehrfädige Prozesortechnik**
 - **Simultaneous Multithreading: Beispiele**
 - Compaq Alpha 21464 (EV8), ursprünglich angekündigt für 2002/2003, Entwicklung aber eingestellt! Entwicklergruppe jetzt bei Intel
 - Intel P4: Hyperthreading
 - Sun Ultra SPARC IV: Chip Multithreading

- **Literatur:**
 - Brinkschulte, U.; Ungerer, T.:
Microcontroller und Mikroprozessoren.
Springer, Heidelberg, 2002: Kap.: 10.4.3

Vergleich von Prozesortechniken



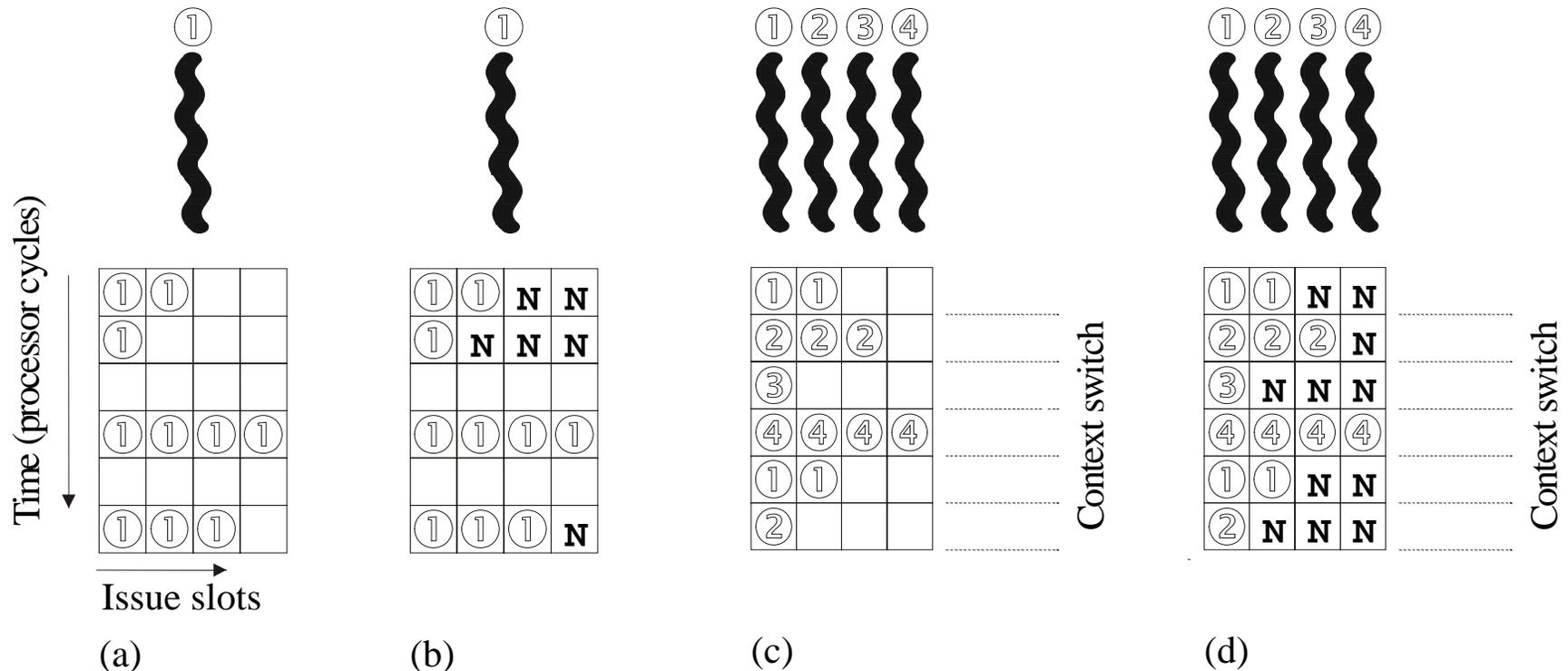
(a): single-threaded scalar

(b) cycle-by-cycle interleaving multithreaded scalar

(c) block interleaving multithreaded scalar

(siehe Brinkschulte, Ungerer: Mikrocontroller und Mikroprozessoren: Kap. 10.4.3)

Vergleich von Prozessortechniken



(a) Superskalare Technik
(b) VLIW-Technik

(c) Cycle-by-cycle Interleaving
(d) Cycle-by-cycle VLIW

(siehe Brinkschulte, Ungerer: Mikrocontroller und Mikroprozessoren: Kap. 10.4.3)

- Architektur und Mikroarchitektur von Prozessoren
 - Architektur
 - RISC-Prinzip: Pipelining
 - Superskalare Prozesortechnik
 - VLIW/EPIC
 - Multithreading
 - Vergleich

Rechnerstrukturen

Vorlesung im Sommersemester 2008

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik



- **Kapitel 3: Multiprozessoren – Parallelismus auf Prozess/Thread-Ebene**

3.1: Motivation

- Überblick

- Allgemeine Grundlagen, parallele Programmierung, Verbindungsstrukturen, Leistungsfähigkeit
- Speichergekoppelte Multiprozessoren: SMP und DSM, Cache-Kohärenz und Speicherkonsistenz, Rechnerbeispiele
- Nachrichtengekoppelte Multiprozessoren, Beispielrechner

- **Einordnung:**
 - Klassifikation nach Flynn: MIMD-Rechner
- **Warum Multiprozessorsysteme?**
 - Hohe Anforderungen von Anwendungen an die Rechenleistung
 - Technisch-wissenschaftlicher Bereich
 - Rechnergestützte Simulation
 - Kommerzieller Bereich
 - Server, Datenbank-Anwendungen, WEB

• Motivation

– Hohe Anforderungen von Anwendungen an die Rechenleistung

• Beispiel technisch-wissenschaftlicher Bereich

- „Grand Challenges“
- Rechnergestützte Simulation
 - Strömungsmechanik
 - Modellierung der globalen klimatischen Veränderungen
 - Evolution von Galaxien
 - Struktur von Materialien
 -

➔ Anforderung an die Rechenleistung: Bereich Tera-, bzw. Petaflop

- Motivation

- Hohe Anforderungen von Anwendungen an die Rechenleistung
- Höchstleistungsrechner:
 - TOP500-Liste
 - Führt die schnellsten Rechner der Welt auf
 - Erscheint immer im Juni und im November eines Jahres
 - <http://www.top500.org>
 - » Beispiel: TOP500 Liste (November 2007)

- Motivation

- Höchstleistungsrechner:

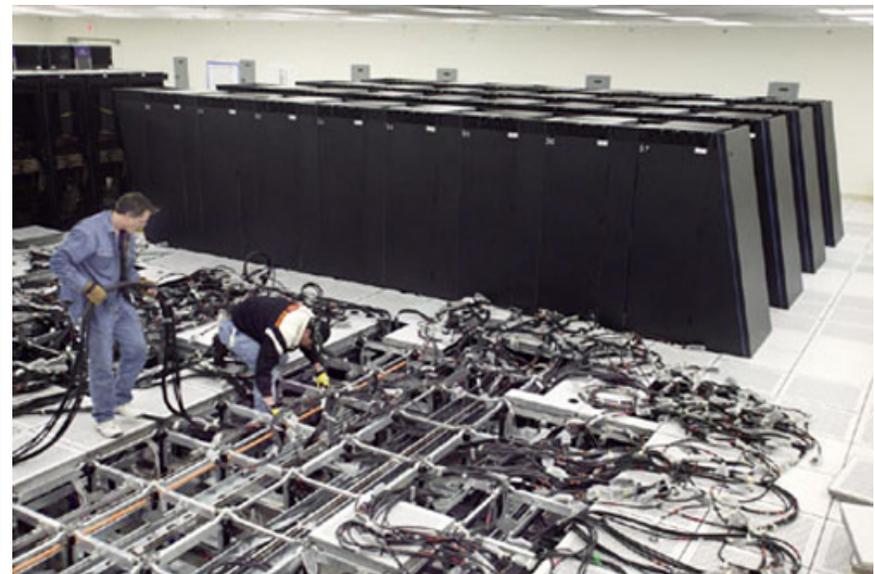
- TOP500-Liste

- Nr.1: BlueGene/L, Modell: eServer Blue Gene Solution
- Standort: DOE/NNSA/LLNL
- Anzahl Prozessoren (PowerPC 440 700 MHz, 2.8 GFlops) : 212992
- Speicher: 73728 GB
- Leistung: 478200 GFLOPS (Linpack)
- Installation: 2007

- Motivation

- Höchstleistungsrechner:

- Nr.1: BlueGene/L, Modell: eServer Blue Gene Solution

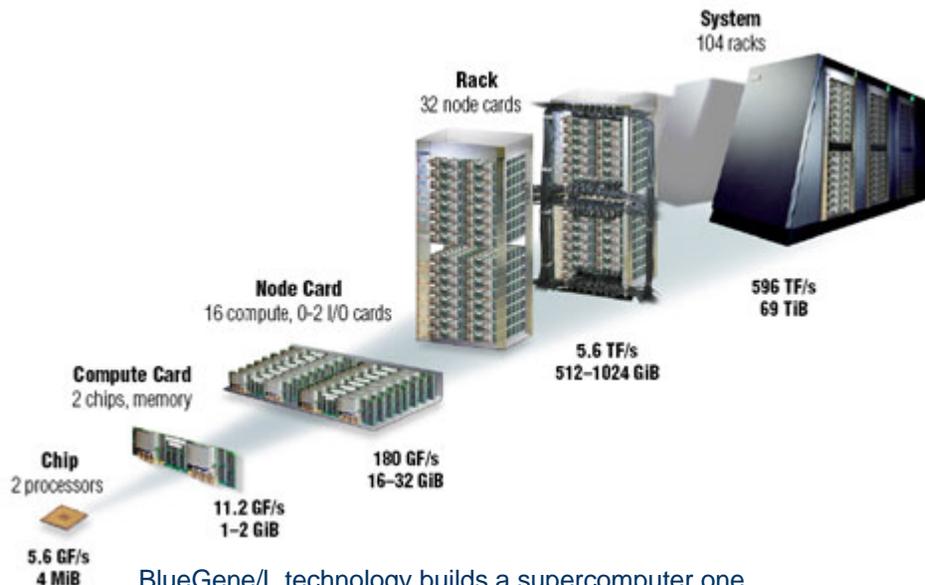


Quelle: https://asc.llnl.gov/computing_resources/bluegenel/photogallery.html

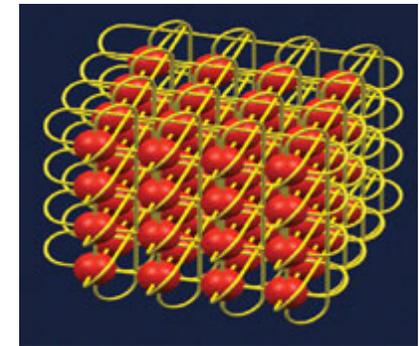
• Motivation

– Höchstleistungsrechner:

- Nr.1: BlueGene/L, Modell: eServer Blue Gene Solution



BlueGene/L technology builds a supercomputer one dual-processor chip at a time. Chips are aggregated into compute cards, which are then assembled into node cards. Each rack holds 2 node cards, and the full machine now comprises 104 racks.



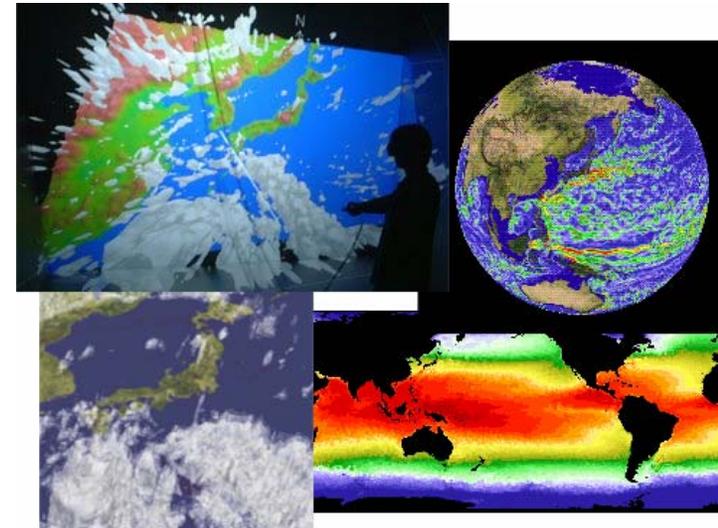
BlueGene/L uses a three-dimensional (3D) torus network in which the nodes (red balls) are connected to their six nearest-neighbor nodes in a 3D mesh. In the torus configuration, the ends of the mesh loop back, thereby eliminating the problem of programming for a mesh with edges. Without these loops, the end nodes would not have six near neighbors.

https://asc.llnl.gov/computing_resources/bluegenel/configuration.html

- Motivation

- Höchstleistungsrechner:

- Earth Simulator (Japan, Platz 30 (TOP500, Nov. 07)
 - Anzahl Prozessoren: 5120
 - Leistung: 35,86 TFLOPS (Linpack),
 - Installation: 2002 (Nummer 1
 - Anwendung: Klimaforschung



Quelle: The Earth Simulator Center;
<http://www.es.jamstec.go.jp/esc/research/Perception/index.en.html>

- Motivation

- Höchstleistungsrechner:

- Earth Simulator

- Ziel des Earth Simulator Project:

- » „The Earth Simulator Project will create a "virtual earth" on a supercomputer to show what the world will look like in the future by means of advanced numerical simulation technology.“
- » „Achievement of high-speed numerical simulations with processing speed of 1000 times higher than that of the most frequently used supercomputers in 1996.“

- Motivation

- Höchstleistungsrechner:

- Earth Simulator

- “Understanding and Prediction of Global Climate Change

- » Occurrence prediction of meteorological disaster
- » Occurrence prediction of El Niño
- » Understanding of effect of global warming
- » Establishment of simulation technology with 1km resolution”

- Motivation

- Höchstleistungsrechner:

- Earth Simulator

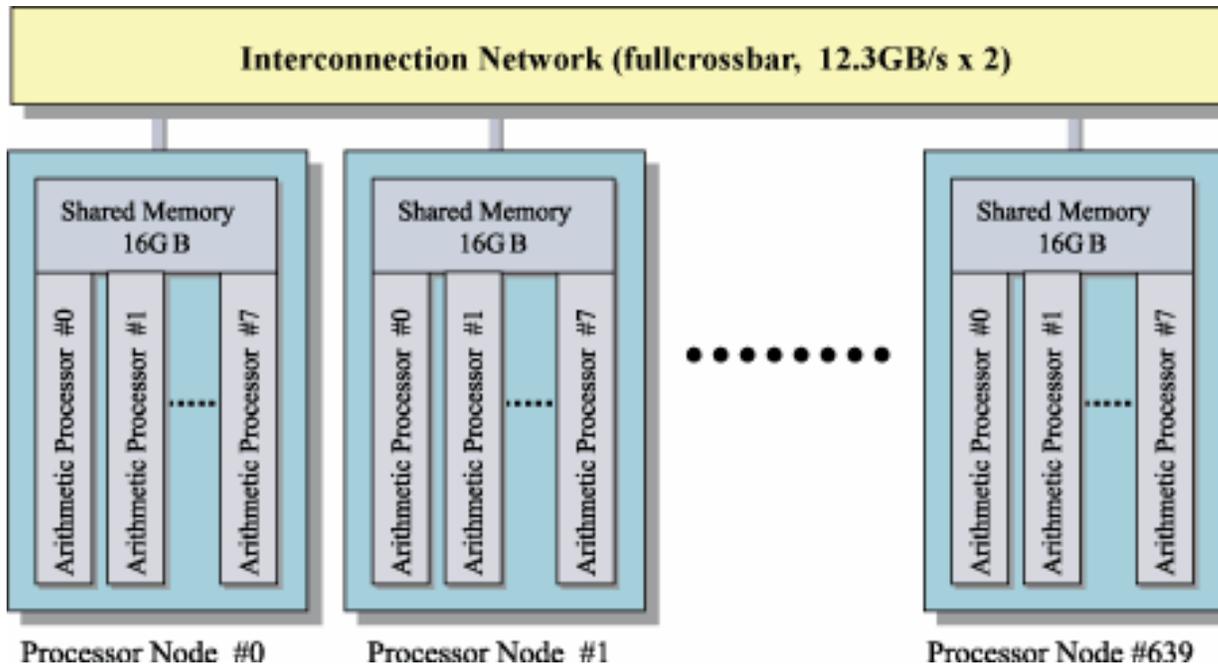
- “Understanding of Plate Tectonics

- » Understanding of long-range crustal movements
- » Understanding of mechanism of seismicity
- » Understanding of migration of underground water and materials transfer in strata”

- Motivation

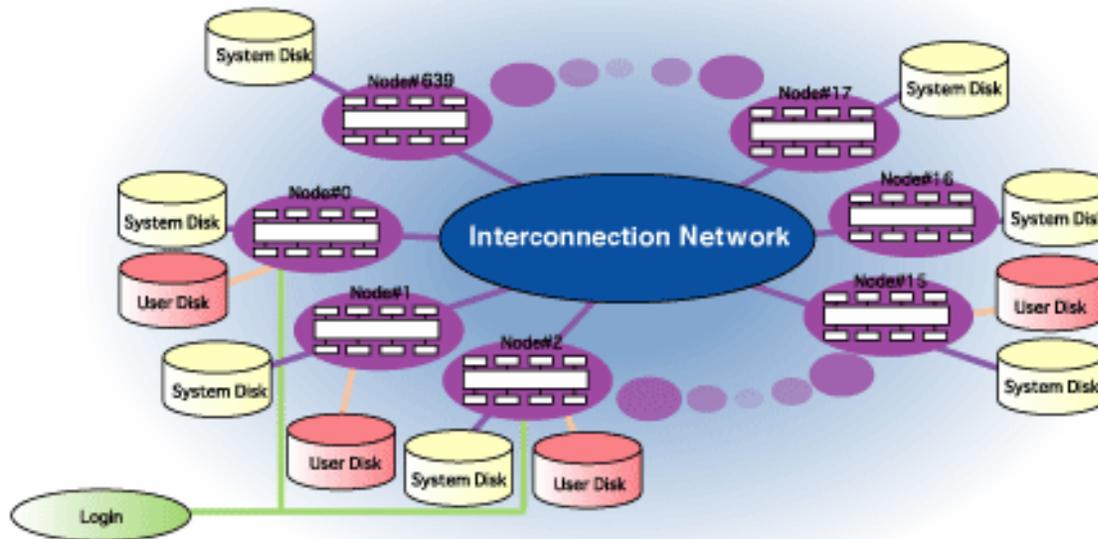
- Höchstleistungsrechner:

- Earth Simulator (Japan)
 - Systemkonfiguration



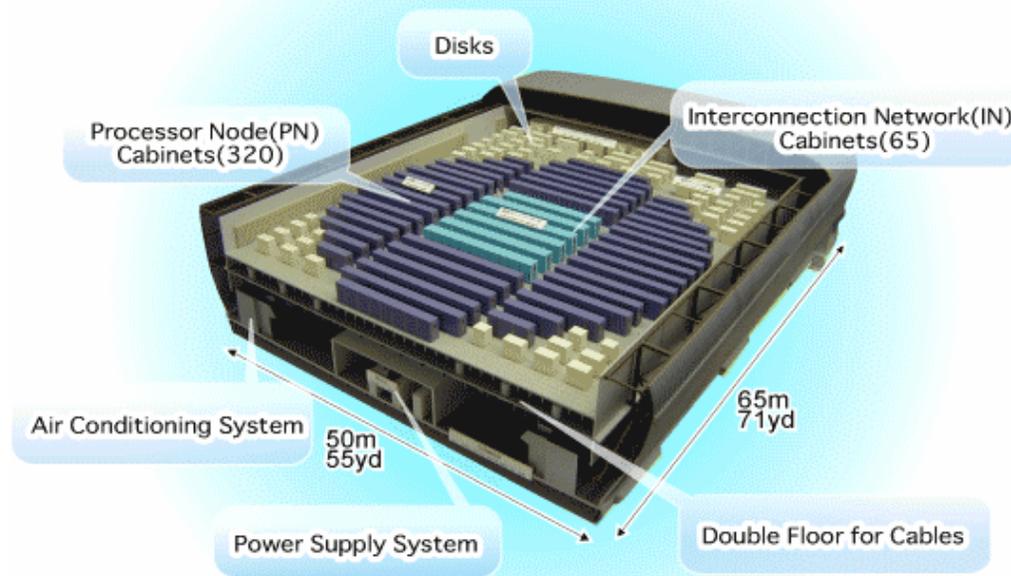
Quelle: The Earth Simulator Center;
<http://www.es.jamstec.go.jp/esc/eng/Hardware/system.html>

- Motivation
 - Höchstleistungsrechner:
 - Earth Simulator (Japan)



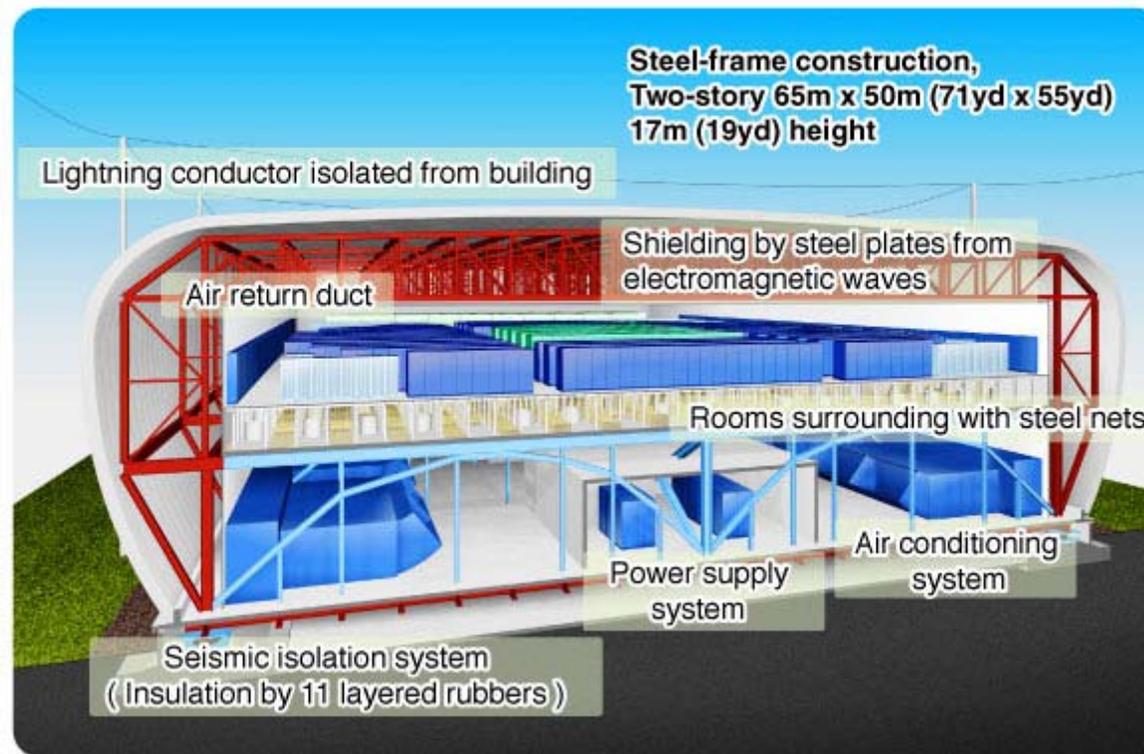
Quelle: The Earth Simulator Center;
<http://www.es.jamstec.go.jp/esc/research/Perception/index.en.html>

- Motivation
 - Höchstleistungsrechner:
 - Earth Simulator (Japan)



Quelle: The Earth Simulator Center;
<http://www.es.jamstec.go.jp/esc/research/Perception/index.en.html>

- Motivation
 - Höchstleistungsrechner:
 - Earth Simulator (Japan)



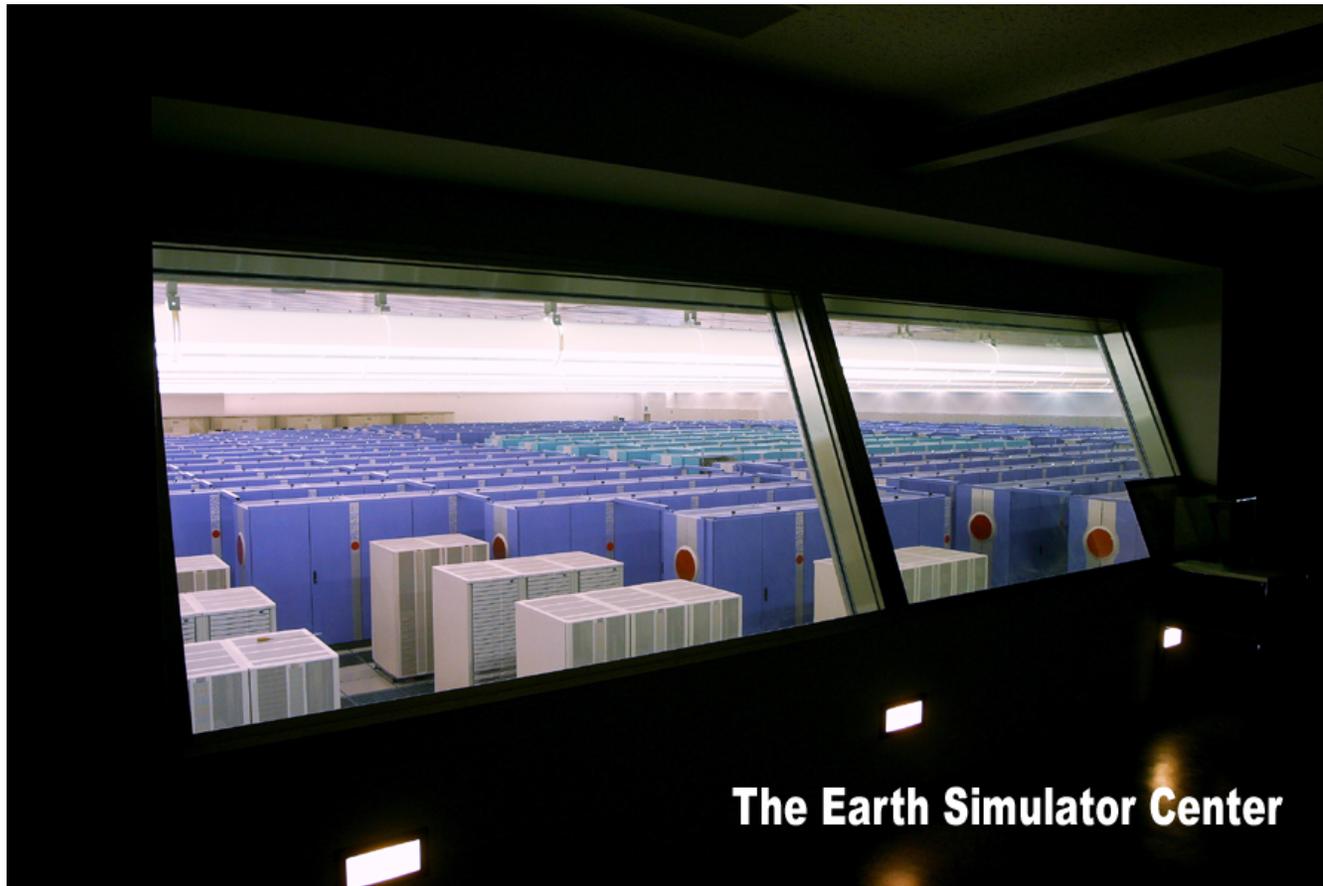
Quelle: The Earth Simulator Center;
<http://www.es.jamstec.go.jp/esc/research/Perception/index.en.html>

- Earth Simulator



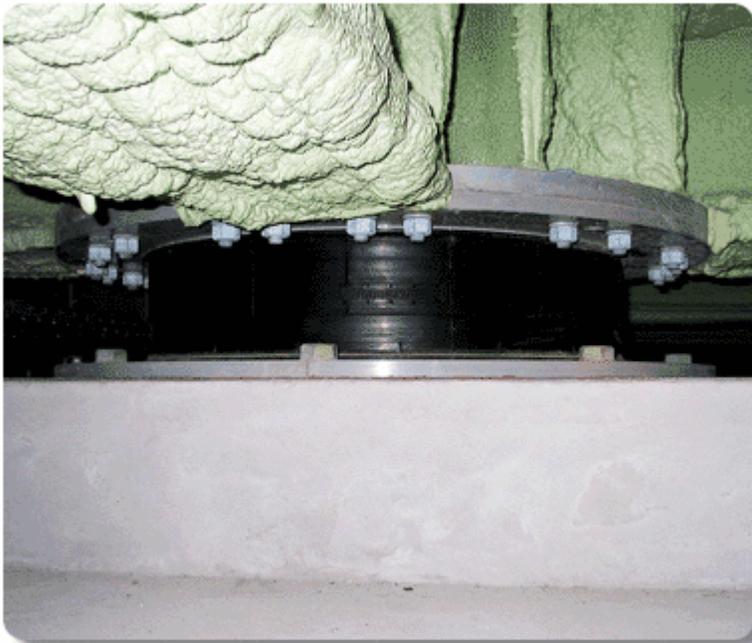
Quelle: The Earth Simulator Center;
<http://www.es.jamstec.go.jp/esc/eng/GC/index.html>

- Earth Simulator



The Earth Simulator Center

- Earth Simulator
 - Erdbebenschutz:

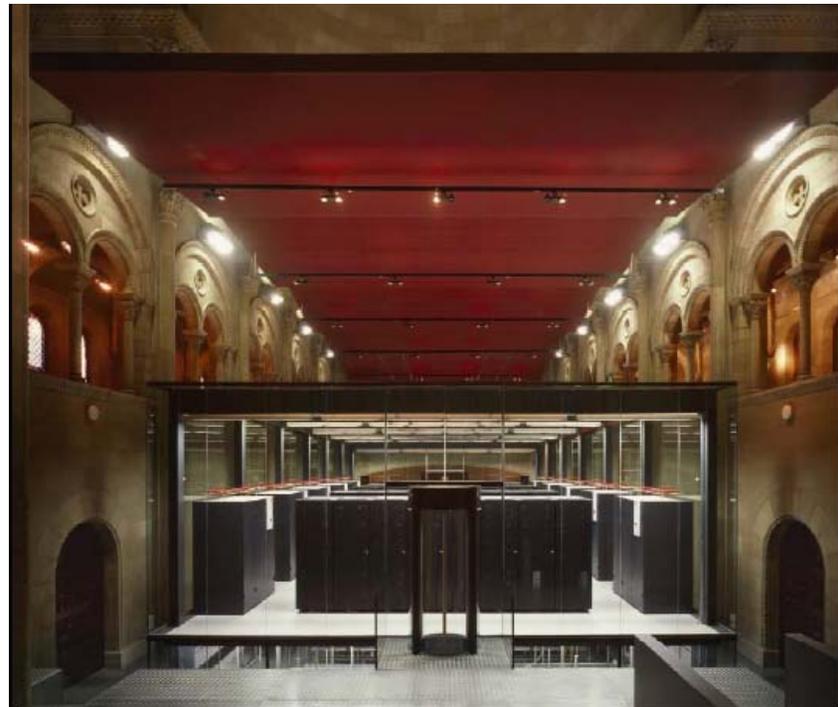


Quelle: The Earth Simulator Center;
<http://www.es.jamstec.go.jp/esc/research/Perception/index.en.html>

- Barcelona Supercomputer Center
 - MareNostrum
 - [BladeCenter JS21 Cluster, PPC 970, 2.3 GHz, Myrinet](#)
 - 10240 Prozessoren



By courtesy of Barcelona Supercomputing Center - www.bsc.es



- **Kapitel 3: Multiprozessoren – Parallelismus auf Prozess/Thread-Ebene**

3.2: Allgemeine Grundlagen

- Parallele Architekturen

- Definition:

- Parallelrechner:

- „A collection of processing elements that communicate and cooperate to solve large problems“ (Almase and Gottlieb, 1989)
- Betrachtung einer parallelen Architektur als eine Erweiterung des Konzepts einer konventionellen Rechnerarchitektur um eine Kommunikationsarchitektur

- **Rechnerarchitektur**

- **Abstraktion**

- Benutzer-/System-Schnittstelle
- Hardware-/Software-Schnittstelle

- **Architektur**

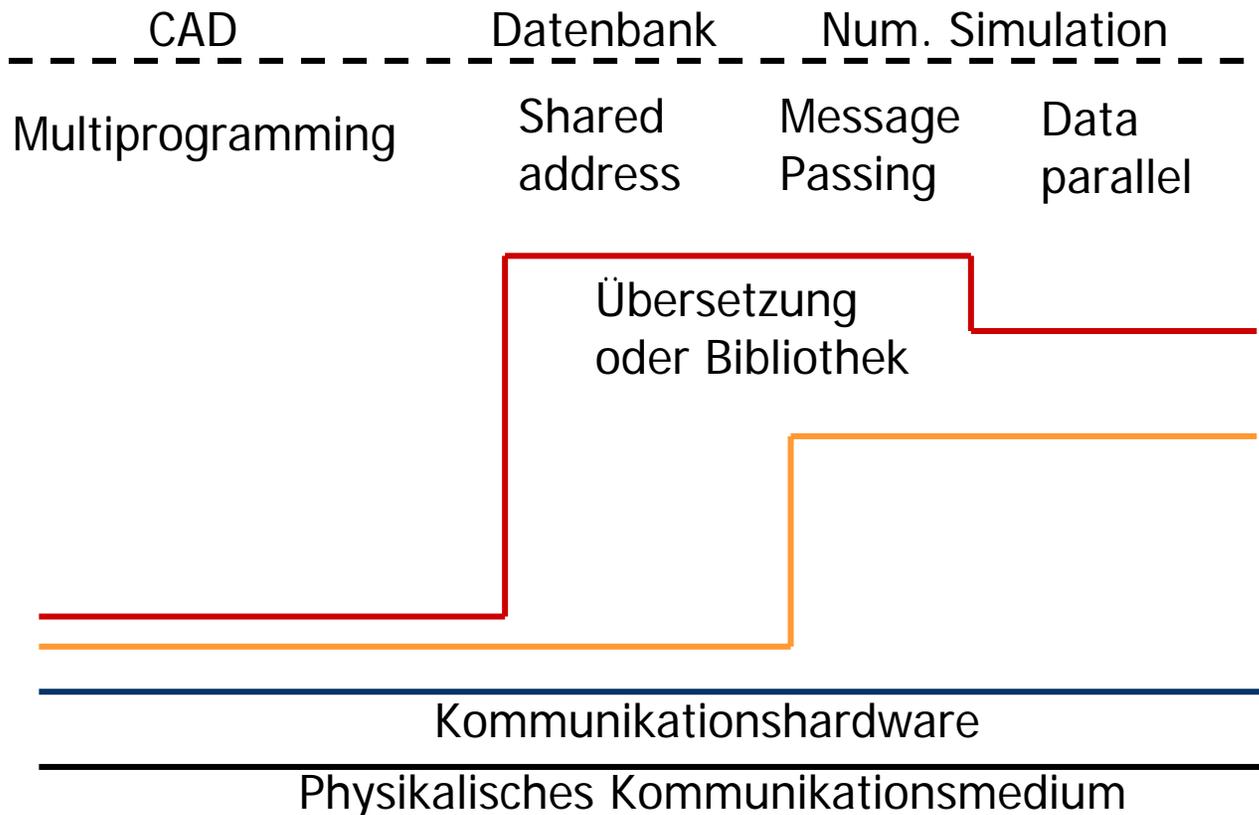
- Spezifiziert die Menge der Operationen an den Schnittstellen und die Datentypen, auf denen diese operieren

- **Organisation**

- Realisierung der Abstraktionen

- **Kommunikationsarchitektur**
 - Abstraktion
 - Benutzer-/System-Schnittstelle
 - Hardware-/Software-Schnittstelle
 - Architektur
 - Spezifiziert die Kommunikations- und Synchronisationsoperationen
 - Organisation
 - Realisierung dieser Operationen

- **Parallele Architekturen**
– Abstraktion



Parallele Anwendung

Programmiermodell

Kommunikations-abstraktion

Benutzer/System-Schnittstelle

Hardware/Software-Schnittstelle

• Parallele Architekturen

– Programmiermodell

- Abstraktion einer parallelen Maschine, auf der der Anwender sein Programm formuliert
- Spezifiziert, wie Teile des Programms parallel abgearbeitet werden, wie Informationen ausgetauscht werden und welche Synchronisationsoperationen verfügbar sind, um die Aktivitäten zu koordinieren
- Anwendungen werden auf der Grundlage eines parallelen Programmiermodells formuliert

- **Parallele Architekturen**
 - Programmiermodell
 - Multiprogramming
 - Menge von unabhängigen sequentiellen Programmen
 - Keine Kommunikation oder Koordination

- **Parallele Architekturen**

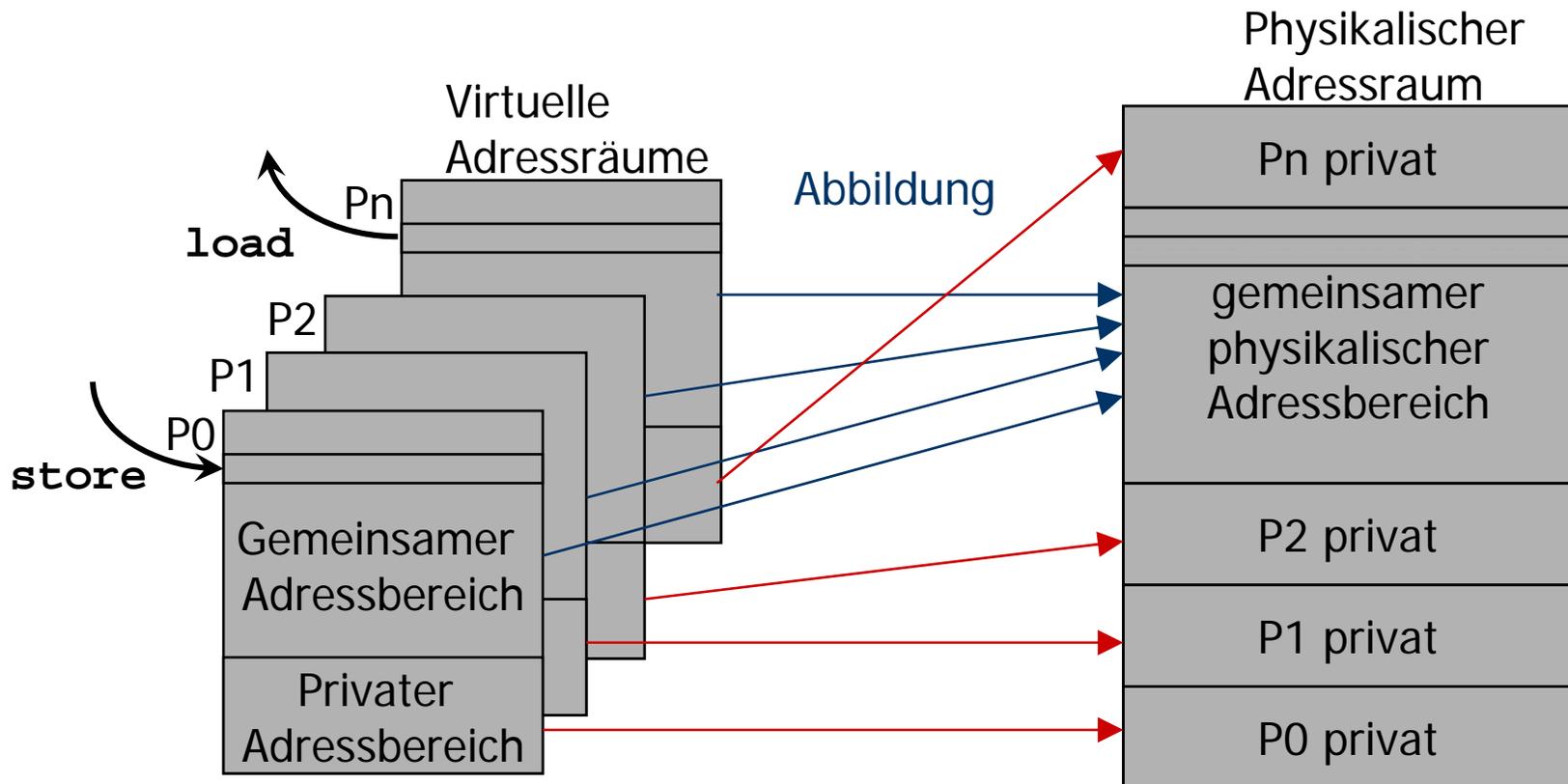
- Programmiermodell

- **Gemeinsamer Speicher (Shared Address Space)**

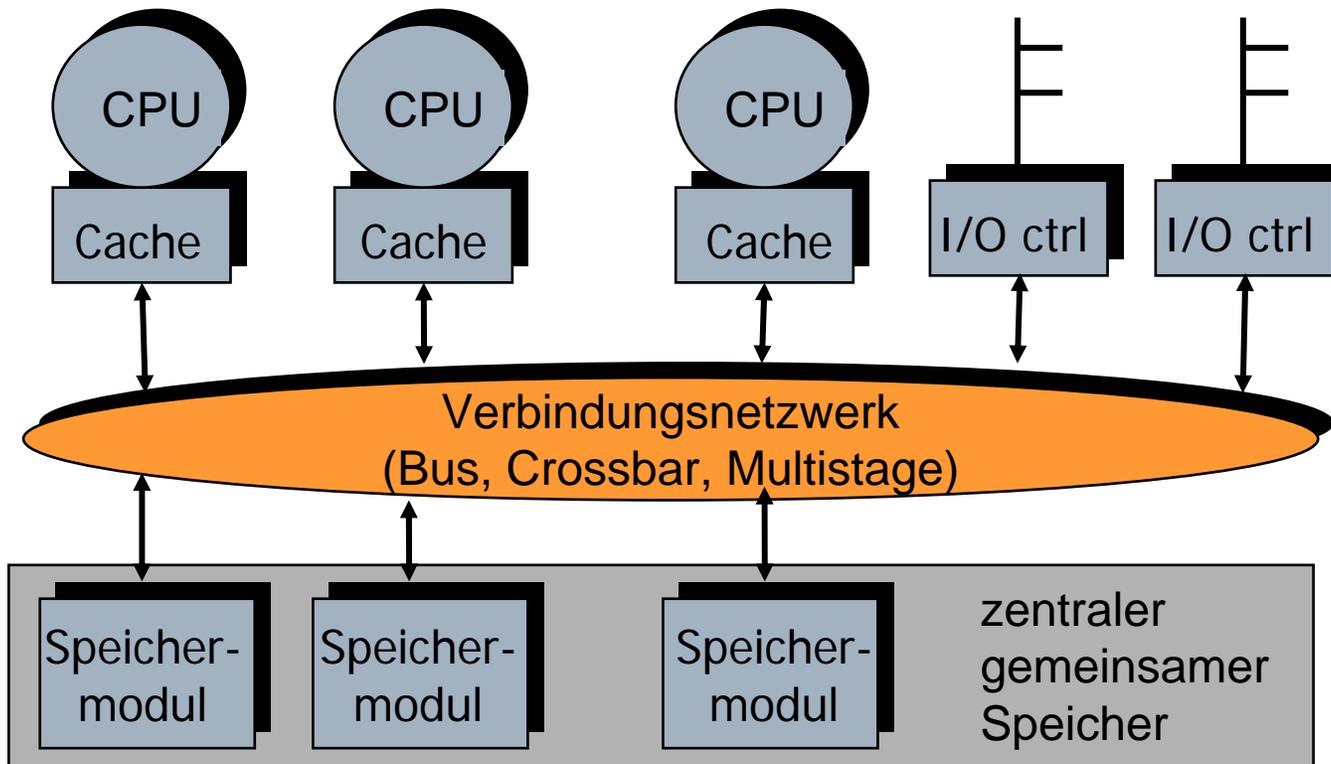
- Kommunikation und Koordination von Prozessen (Threads) über gemeinsame Variablen und Zeiger, die gemeinsame Adressen referenzieren
- Kommunikationsarchitektur
 - » Verwendung konventioneller Speicheroperationen für die Kommunikation über gemeinsame Adressen
 - » Atomare Synchronisationsoperationen

• Parallele Architekturen

- Gemeinsamer Speicher (Shared Address Space)



- **Parallele Architekturen**
 - Multiprozessor mit gemeinsamem Speicher



- **Parallele Architekturen**

- Programmiermodell

- Nachrichtenorientiertes Programmiermodell (Message Passing)

- Kommunikation der Prozesse (Threads) mit Hilfe von Nachrichten

- » Kein gemeinsamer Adressbereich

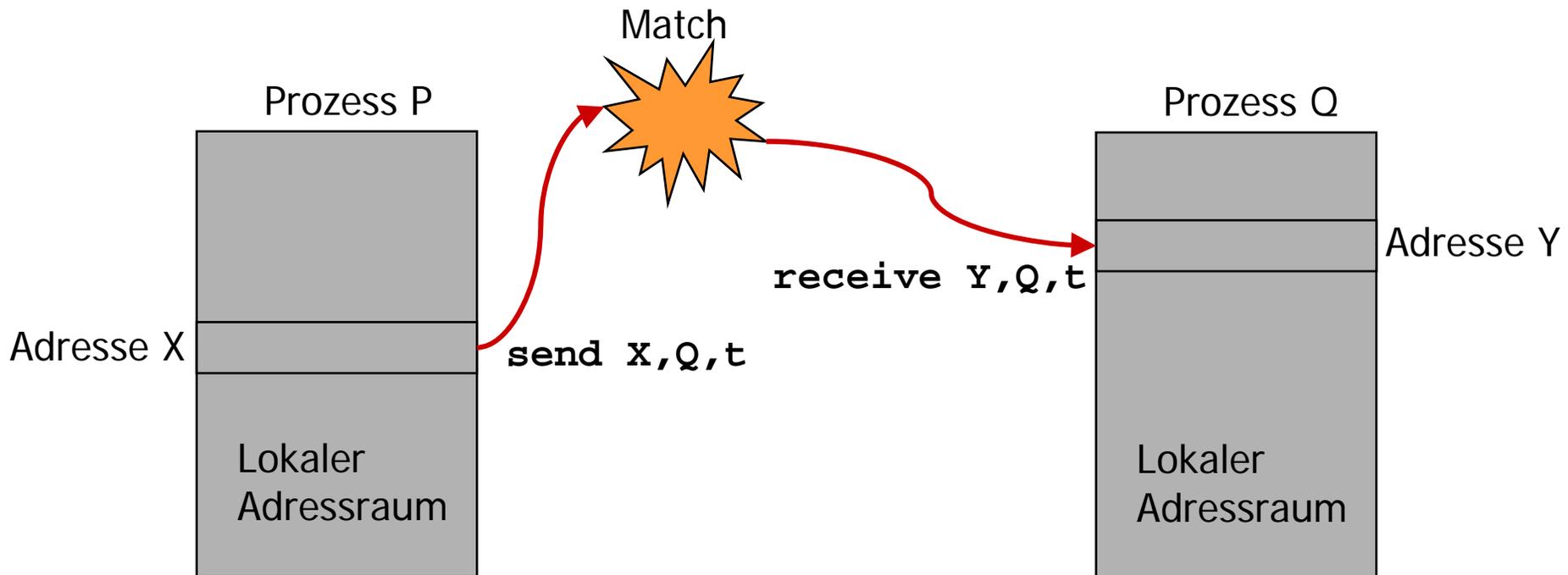
- Kommunikationsarchitektur

- » Verwendung von korrespondierenden Send- und Receive-Operationen

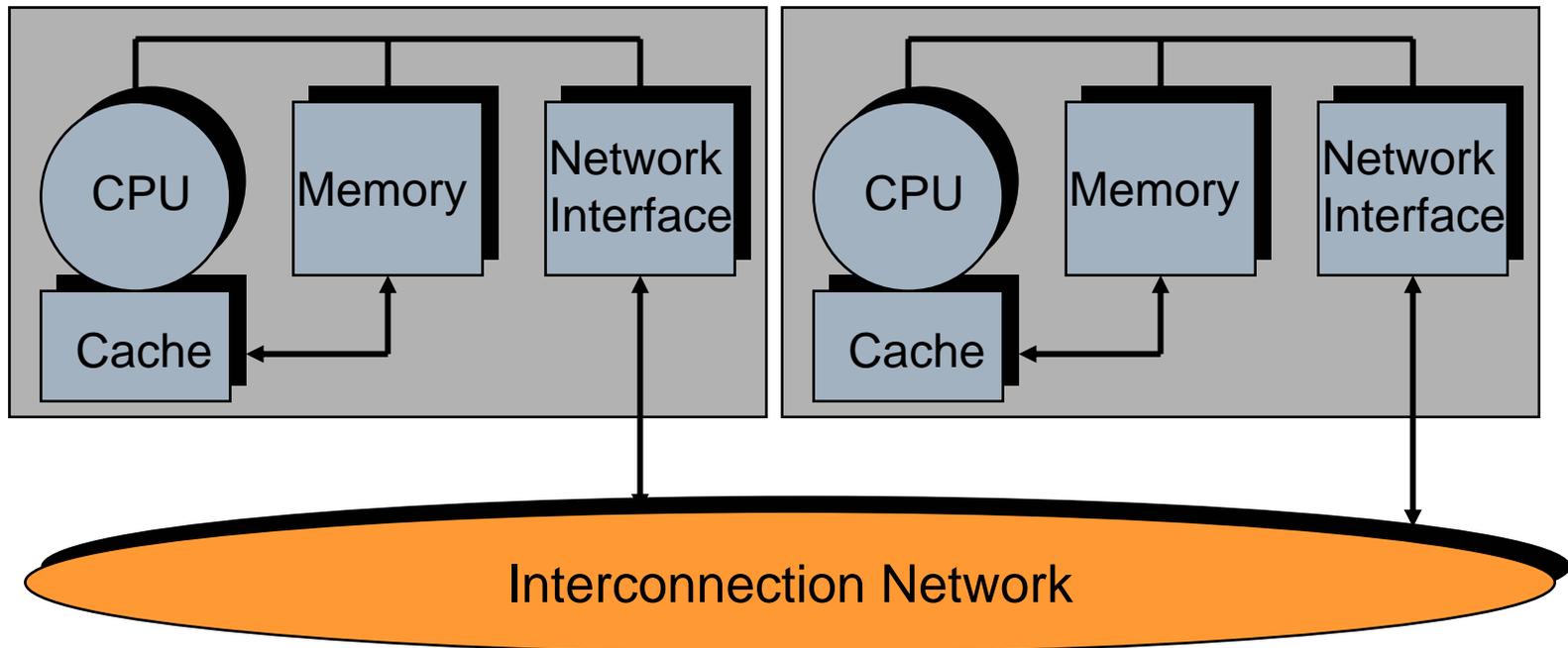
- » Send: Spezifikation eines lokalen Datenpuffers und eines Empfangsprozesses (auf einem entfernten Prozessor)

- » Receive: Spezifikation des Sende-Prozesses und eines lokalen Datenpuffers, in den die Daten ankommen

- **Parallele Architekturen**
 - Nachrichtenorientiertes Programmiermodell (Message Passing)



- **Parallele Architekturen**
 - Multiprozessor mit verteiltem Speicher



- **Parallele Architekturen**
 - Programmiermodell
 - Datenparallelismus
 - Gleichzeitige Ausführung von Operationen auf getrennten Elementen einer Datenmenge (Feld, Matrix)
 - Typischerweise in Vektorprozessoren