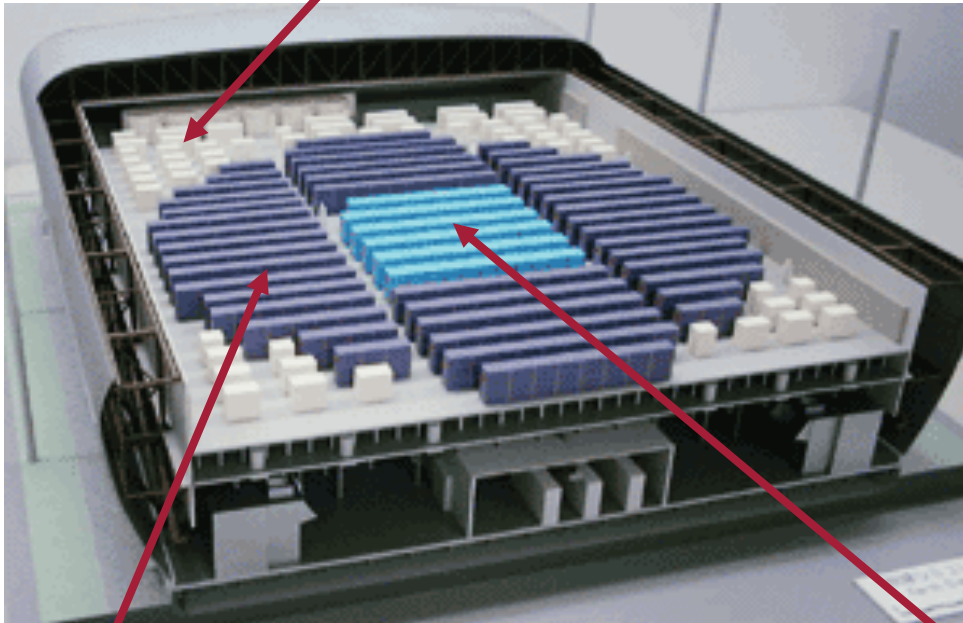

1. Parallelverarbeitung

Supercomputer (Beispiel Earth Simulator, 2001)

Disk Array



CPUs

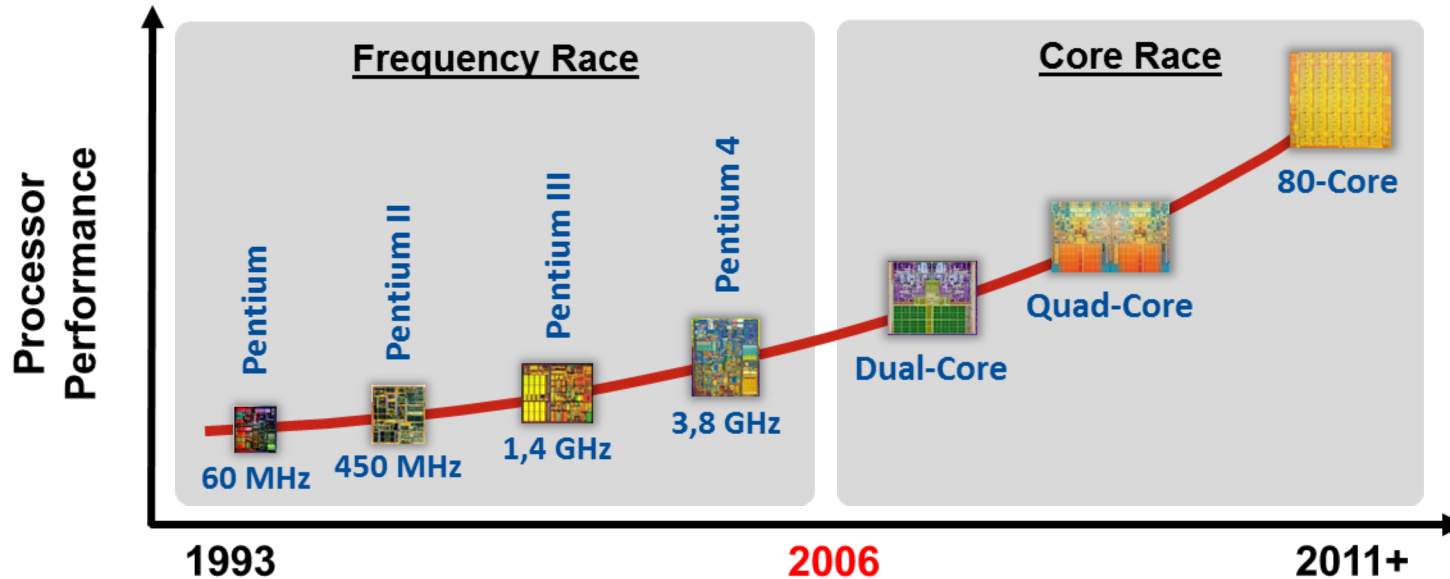


Crossbar Switch

Typische Probleme im Supercomputing

- **Grand Challenges:** Probleme, die ohne paralleles Rechnen in „vernünftiger Zeit“ nicht lösbar sind.
 - Klimaforschung
 - Erdbebensimulation
 - Simulation von Galaxien
 - Rechenintensive Probleme mit „kurzer“ Deadline
 - Wettervorhersage
 - Simulationen im Entwicklungsprozess eines Produkts, z.B. virtuelle Crash-Tests bei Automobilentwicklung
 - Anwendungen mit Realzeitanforderungen, z.B. virtuelle Realität
-

Multi-Core CPUs



- **Prinzip:** Es werden mehrere vollständige Prozessoren (= Cores) auf einem Chip integriert
 - Anzahl der Cores verdoppelt sich ca. alle 2 Jahre (gem. Moore's Law.)
- Heutzutage allgegenwärtig, da andere Methoden der Leistungssteigerung (z.B. höhere Taktfrequenz) nicht mehr greifen.
- **Nicht parallelisierte Programme können nur einen Bruchteil der Leistungsfähigkeit nutzen (z.B. 25 % bei Quadcore)**

Parallelität in Rechnersystemen

Parallelität tritt auf unterschiedlichen Ebenen auf:

- **Bitebenenparallelität:** Datenbits werden zu Datenworten zusammengefasst und parallel verarbeitet.
- **Befehlsebenenparallelität:** Maschinenbefehle werden implizit parallel ausgeführt.
 - Pipelining: Überlappung der einzelnen Phasen der Befehlsausführung.
 - Superskalere Architektur: Betrieb mehrerer Pipelines.
- **Programmebenenparallelität:**
 - Parallelität wird durch geeignete Programmierkonstrukte oder Tools explizit im Programm festgelegt.

Exkurs: Parallelität auf Befehlsebene

- Pipeline-Architektur: Überlappende Ausführung der Verarbeitungsphasen eines Maschinenbefehls:

- Instruction Fetch (IF)
- Instruction Decode (ID)
- Execute (EX)
- Memory Access (MA)
- Write Back (WB)



- Ziel: Pro Takt wird ein Befehl abgeschlossen.
 - Superskalare Architektur: Mehrere Pipelines (bzw. einzelne Stufen einer Pipeline) werden parallel betrieben.
 - Ziel: Pro Takt werden mehrere Befehle abgeschlossen.
-

Programmebenenparallelität

- **Bit- und Befehlsebenenparallelität** finden heute breite Verwendung.
 - Warum macht Programmebenenparallelität Sinn?
 - **Performance:** Steigerung der absoluten Rechenleistung durch Verwendung mehrerer Prozessoren wird möglich.
 - **Software-Engineering:** Einfachere Programmerstellung durch Ausnutzung natürlicher Parallelität der Anwendung.
 - **Paralleles Rechnen:** Beschleunigung der Berechnung eines Problems durch den Einsatz **mehrerer Prozessoren** und **Programmebenenparallelität**.
-

Wichtige Metriken der Parallelverarbeitung

- **Sequentielle Laufzeit T_s** : Zeit, die zwischen dem Programmstart und dem Programmende bei der Ausführung auf einem sequentiellen Rechner verstreicht.
- **Parallele Laufzeit T_p** : Zeit zwischen dem Start und dem Ende der parallelen Programmausführung auf p Prozessoren.
- **Speedup: $S := T_s / T_p$**
 - Maß für die durch Parallelverarbeitung erzielte Beschleunigung
- **Effizienz: $E := S / p$**
 - Maß für den Ausnutzungsgrad des Parallelrechners