Erfahrungen und Benchmarks mit Dual-Core-Prozessoren

Georg Hager
Regionales Rechenzentrum Erlangen (RRZE)

ZKI AK Supercomputing
Karlsruhe, 22./23.09.2005

Dual Core: Anbieter heute

- **IBM Power4/Power5 (Power5 mit SMT)**
  - **p6xx: MCM-basiert**
  - **OpenPower 7xx: DCM-basiert, Struktur ähnlich Dual-DC Opterons**
  - **System im Test: OpenPower720 mit 2 DC Power5 @ 1.65 GHz und 10GB Speicher**
- **AMD Opteron (seit August auch Athlon64)**
  - **diverse Anbieter, Upgrade von Sockel939-Systemen u.U. möglich**
  - **bis zu 4, bald 8 Sockets pro Board**
  - **Systeme im Test: Sun Fire V20z (2x2 Cores), Transtec-Testsystem (4x2 Cores)**
- **Intel Pentium D**
  - **single Socket, FSB800**
  - **Hyperthreading**
  - **System im Test: FSC "Esprimo" PC, 2.8 GHz**
- **Sun UltraSparc IV+**

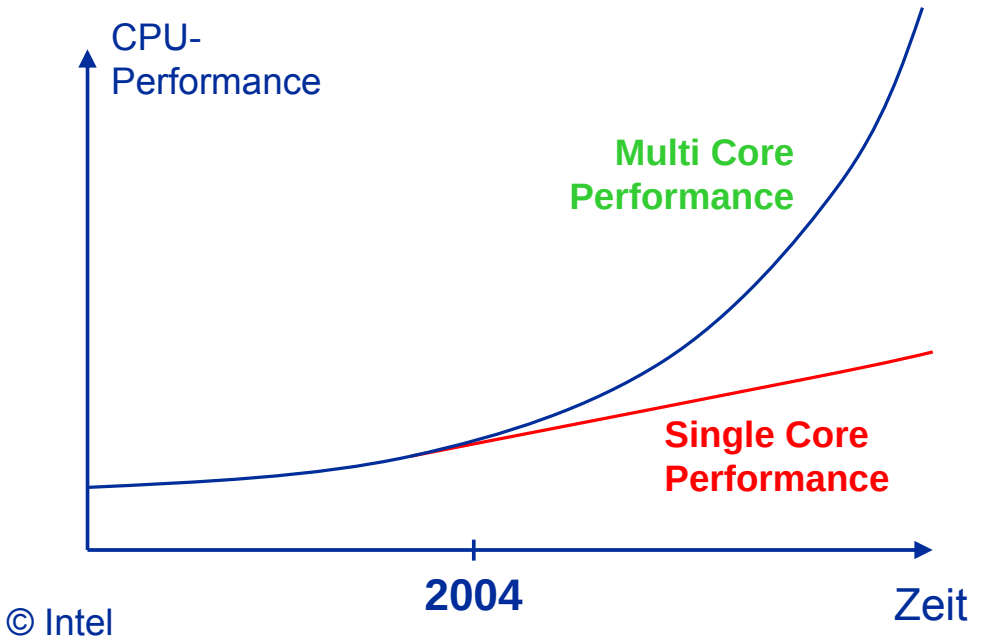
Dual Core: Die wichtigsten Fragen

- Was bringt der zweite Kern?
- Geringere Taktfrequenz = schwächere Performance?
- Geteilte Speicherbandbreite = halbe Performance?
- Können 2 Cores die Bandbreite besser nutzen als einer?
- Getrennte/gemeinsame Caches: Was ist besser?
- Direkte Kommunikation zwischen den Cores in einem Socket?
- Thread-/Prozess-Placement: Wie werden im halb vollen System die Threads platziert?
- Bei Dual-Core-Multi-Socket-Systemen: Wie erfolgt die Zuordnung von Core zu Speicherkanal?

- Vorläufiges Fazit: Benchmarks sind unbedingt notwendig!

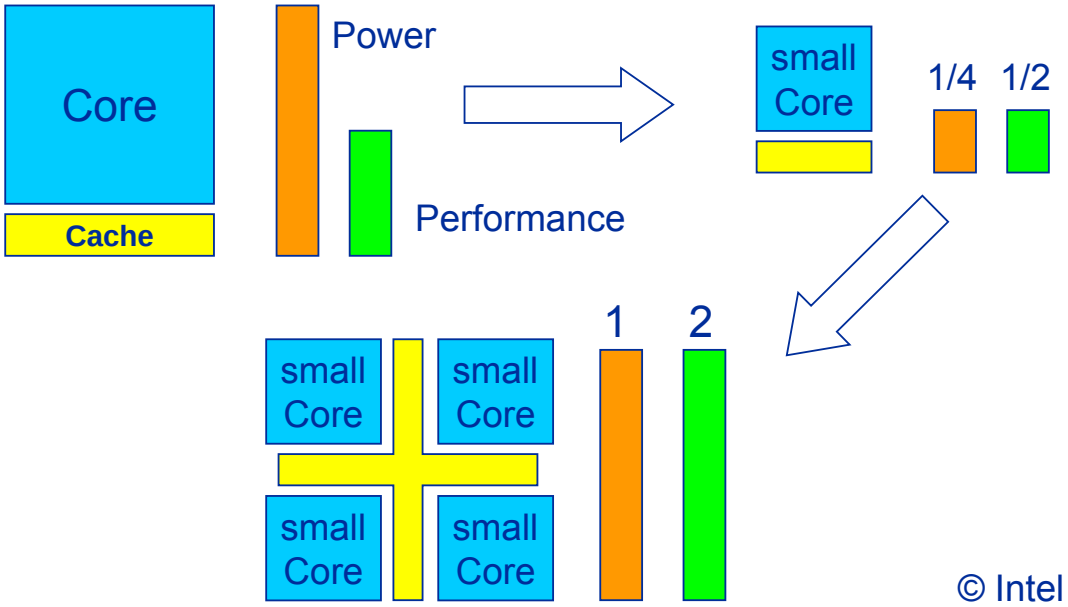
Warum Dual Core?

- Das „Megaflop-Rennen“ wird bald zu Stillstand kommen



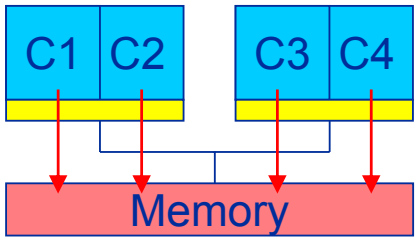
Die Annahmen hinter Multi-Core

- **Mehrere Cores sollen mehr Performance pro Watt auf gleichem Raum bringen**



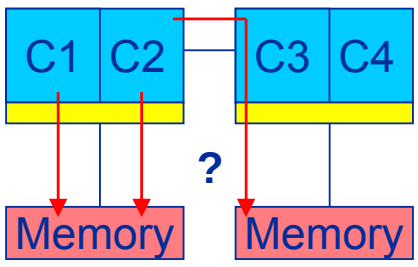
Komplikationen

- **Systeme mit einem Speicherkanal**



Thread/Memory-Zuordnung egal!

- **Systeme mit mehreren Speicherkanälen (ccNUMA)**

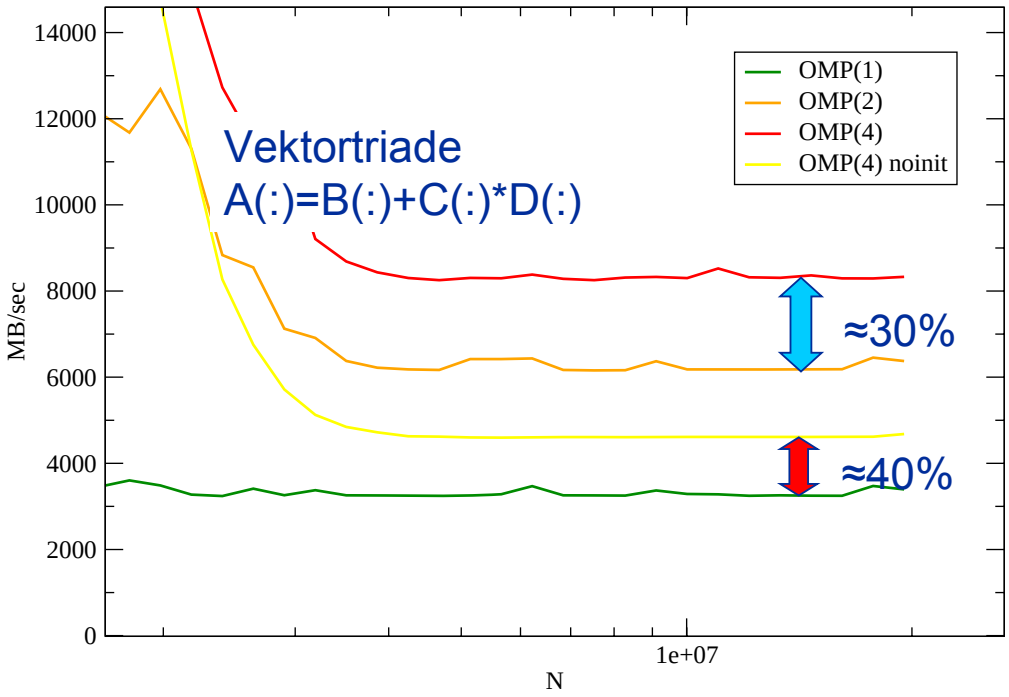


Thread/Memory-Zuordnung essenziell!

Thread/Memory Placement wie?

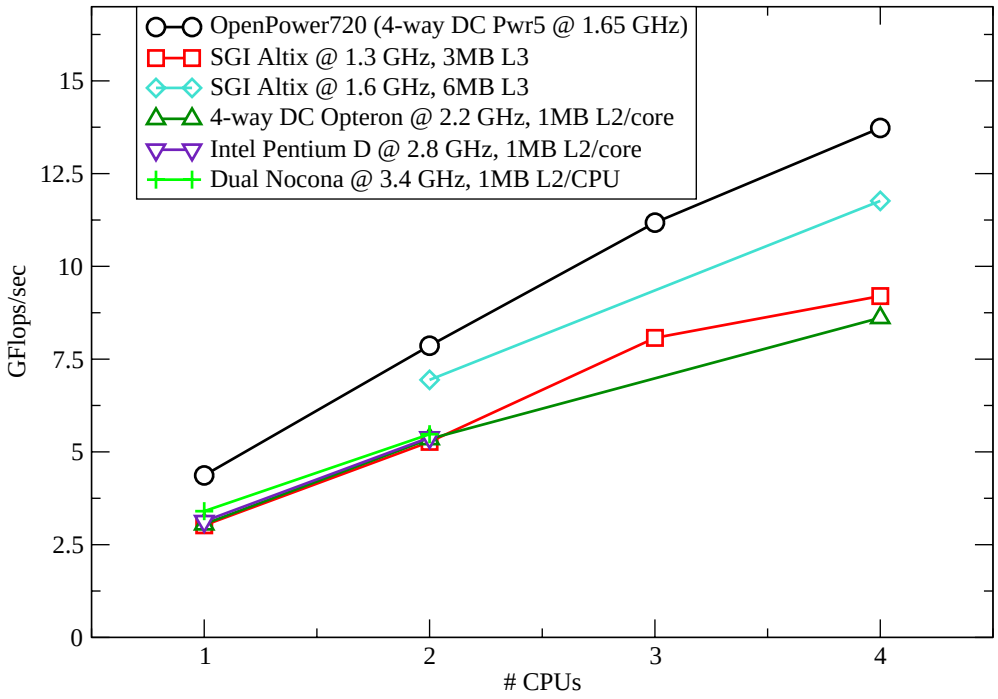
- **Betriebssystem**
  - Automatisches Placement wird besser, ist aber oft nicht optimal
  - Auch bei single-Thread-Programmen kann das Resultat überraschend sein…
- **Compiler**
  - Pathscale-Compiler hat differenzierte Thread-Placement-Mechanismen eingebaut
- **Benutzer (taskset / numactl)**
  - Nur sinnvoll, wenn man den Knoten für sich hat
    ```
    taskset -c 0,2 numactl -m 0,1 ./a.out
    ```
  - Integration in MPI-Umgebung nicht trivial
- **Programm (first touch)**
  - OpenMP: Initialisierung der Daten muss parallelisiert werden wie die Kernschleifen des Programmes – immer möglich?

Speicherperformance am Beispiel IBM OpenPower720



Dual vs. Single Core:

DMRG Benchmark (OpenMP, DGEMM-lastig)



Weitere Komplikationen

- **Beim Übergang von Single nach Dual Core verringert sich in einem Cluster die Bisektionsbandbreite pro CPU um den Faktor 2**
- **Fragen:**
  - **Ist dieser Effekt wichtig für die Applikationen?**
  - **Wie viele Cores brauche ich, um die Bandbreite eines Netzwerkadapters zu sättigen?**
  - **Ist die Netzwerkbandbreite mit der gegebenen MPI-Implementierung überhaupt ein Bottleneck?**

Resultate

- **Bringt der zweite Core etwas?**
  - **Die Frage müsste lauten: Wenn mein Applikationsmix durchschnittlich X% Performancegewinn hat und das DC-System Y% teurer ist als das SC-System, ist X>Y?**
  - **Bekommt man die Placement-Probleme bei NUMA-Systemen in den Griff?**
  - **Haben wir überhaupt eine Wahl?**
- **I.A. kann der zweite Core aus dem Speicherinterface noch etwas mehr „rauskitzeln“**
  - **OpenPower: 30-40%**
  - **Opteron: 25%**
  - **Pentium D: 0%**
- **Unterschiede in der Taktfrequenz werden durch andere Effekte oft verwischt**
- **Achtung bei Netzwerk-Bisektionsbandbreiten**