Script generated by TTT

Title: Westermann: Informatik_1 (11.01.2012)

Date: Wed Jan 11 14:24:21 CET 2012

Duration: 64:53 min

Pages: 24

Klausuren

- Einführung in die Informatik 1
  Sa 25.2.2012, 9:00 bis 11:00

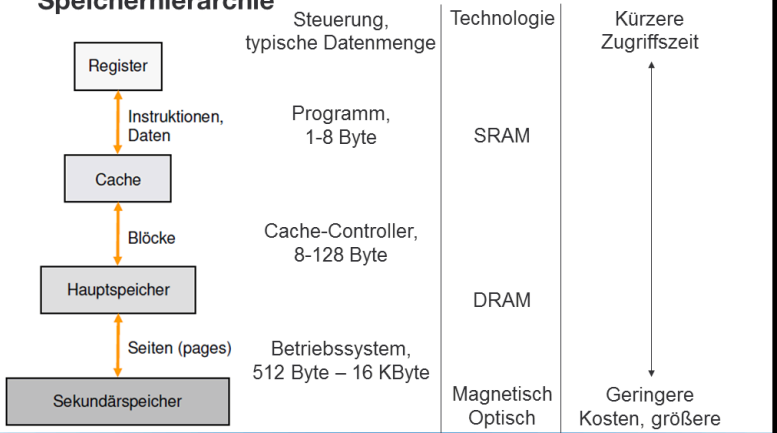
  Wiederholungsklausur: Fr 13.4.2012, 11:00 bis 13:00

- Praktikum Grundlagen der Programmierung:
  Sa 3.3.2012, 12:00 bis 15:00

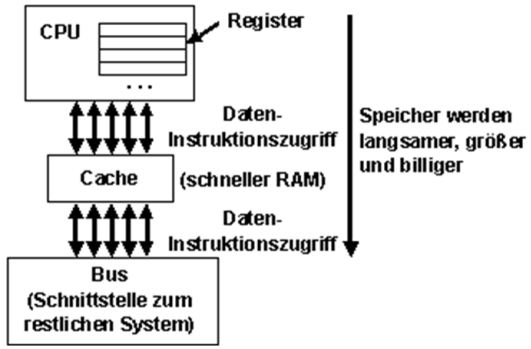
  Wiederholungsklausur: Mo 2.4.2012, 10:30 bis 13:30

- Anmeldung über tumonline!

Speicherhierarchie



Speicherhierarchie



http://www.it-infothek.de/fhtw/ra_bs_02.html

Technologie

- RAM bezeichnet einen Speichertyp dessen Speicherzellen über ihre Speicheradressen direkt, einzeln und "wahlfrei" angesprochen werden können
- SRAM ist statisch, d.h. dass der Speicherinhalt bei anliegender Spannung erhalten bleibt
- DRAM ist dynamisch, d.h. dass der Speicherzustand immer wieder neu (periodisch) aktualisiert werden muss (Refresh)
- Festplatten bestehen aus beweglichen Schreib/Lese-Köpfen, die über die Magnetplatten gleiten (Positionierung) und das Magnetfeld ändern bzw. lesen

Technische Universität München

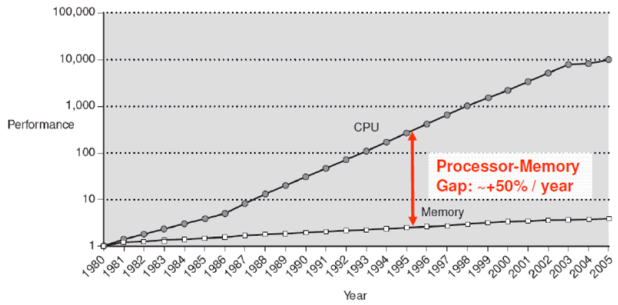
Technologie

- RAM bezeichnet einen Speichertyp dessen Speicherzellen über ihre Speicheradressen direkt, einzeln und "wahlfrei" angesprochen werden können
- SRAM ist statisch, d.h. dass der Speicherinhalt bei anliegender Spannung erhalten bleibt
- DRAM ist dynamisch, d.h. dass der Speicherzustand immer wieder neu (periodisch) aktualisiert werden muss (Refresh)
- Festplatten bestehen aus beweglichen Schreib/Lese-Köpfen, die über die Magnetplatten gleiten (Positionierung) und das Magnetfeld ändern bzw. lesen

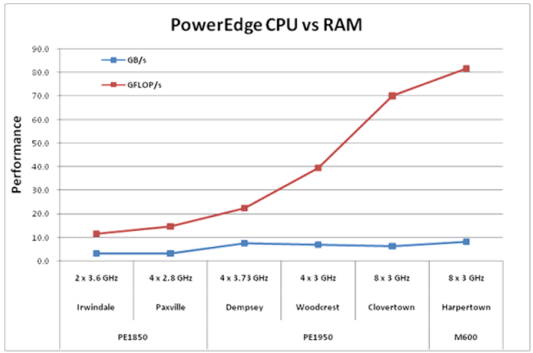
Fakultät für Informatik, Einführung in die Informatik, WS 2011/12
5

Technologische Trends

- Entwicklung der Performance von CPUs und RAM relativ zur Performance von RAM 1980

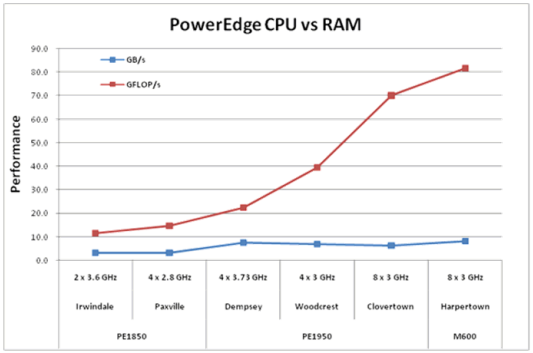


Technologische Trends



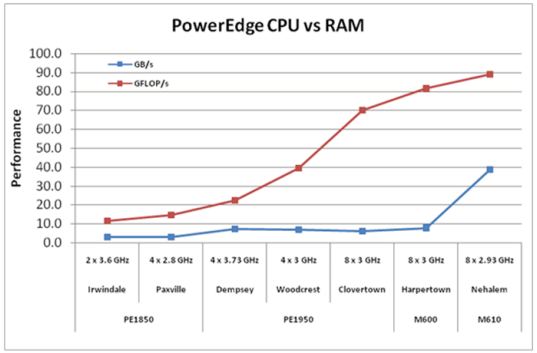
Dreimalige Verdoppelung der CPU Performanz über die letzten 5 Jahre; einmalige Verdoppelung der Speicher Performanz

Technologische Trends



Dreimalige Verdoppelung der CPU Performanz über die letzten 5 Jahre; einmalige Verdoppelung der Speicher Performanz

Technologische Trends



Neue Speicherschnittstellen und schnellerer Speicher verringern Divergenz; aber die sog. “Memory-Wall” wird mit großer Wahrscheinlichkeit bleiben

Technologische Trends

- Fazit:
  Speicherzugriff ist und wird zunehmend der limitierende Faktor bzgl. der Performanz eines Programmes

  Datenintensive Anwendungen benötigen spezielle Datenstrukturen, Datenkompressionsverfahren und Datenzugriffsmechanismen zur Steigerung der Performanz

Technologische Trends

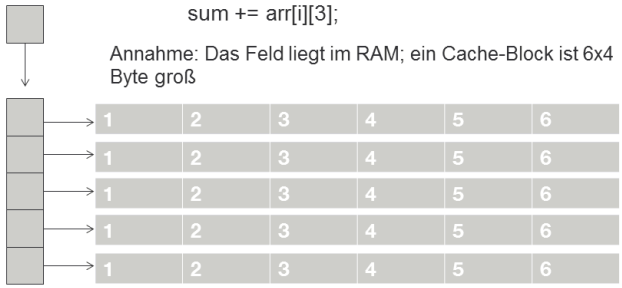
- Fazit:
  Speicherzugriff ist und wird zu d
  - Daten werden nu chen benachb Eb ag
    st eine Dateneinheit auf einer Ebene vorhanden, muss sie auch auf n tieferen Ebenen vorhanden sein.
  - Sind die angeforderten Daten auf einer Ebene verfügbar, dann ist der Zu erfolgreich und es kommt zu einem Hit (Treffer)
  - Sind die Daten nicht verfügbar, dann kommt es zu einem Miss (Fehlz gr die Daten müssen aus der nächsten Ebene geholt werden

Speicherhierarchie

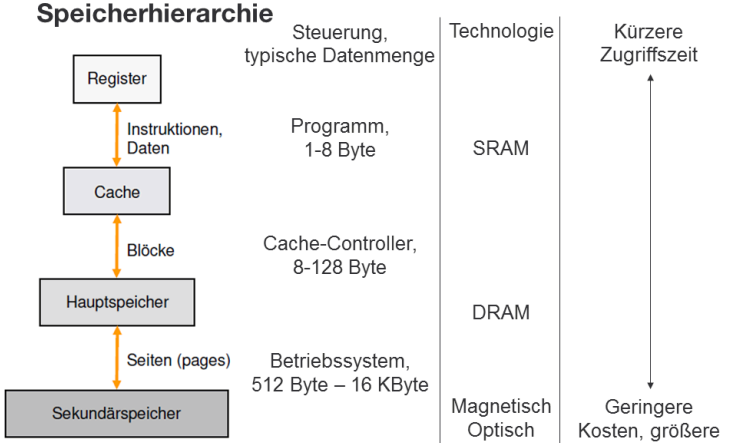
- Beispiel:

```
int arr[5][6] = {{…}…};  // initialization
int sum = 0;
for (int i = 0; i < 5; ++i)
        sum += arr[i][3];
```

Annahme: Das Feld liegt im RAM; ein Cache-Block ist 6x4 Byte groß



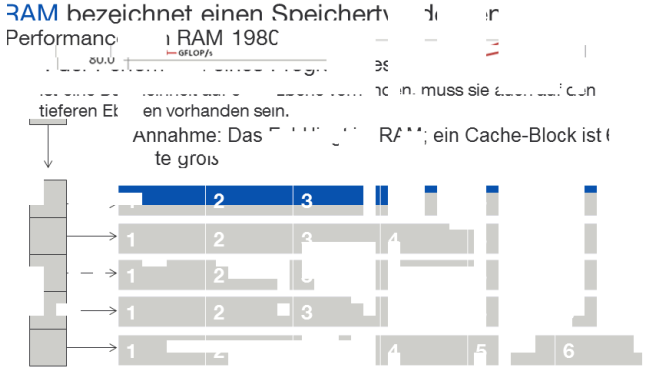
Speicherhierarchie



Speicherhierarchie

RAM bezeichnet einen Speichertyp

Performanc… RAM 1980



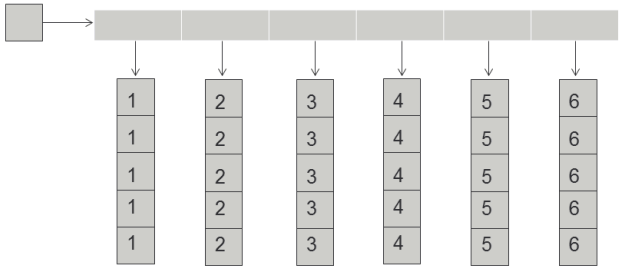
tieferen Eb… …muss sie auch auf den … en vorhanden sein.

Annahme: Das … RAM; ein Cache-Block ist … te groß

Speicherhierarchie

- Beispiel:

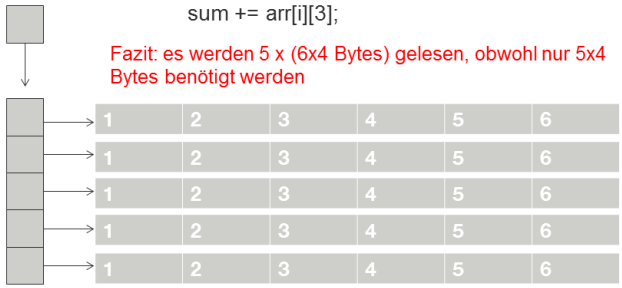
```
int arr[6][5] = {{…}…};  // initialization
int sum = 0;
for (int i = 0; i < 5; ++i)
        sum += arr[3][i];
```



Speicherhierarchie

```
int arr[5][6] = {{…}…};  // initialization
int sum = 0;
for (int i = 0; i < 5; ++i)
        sum += arr[i][3];
```

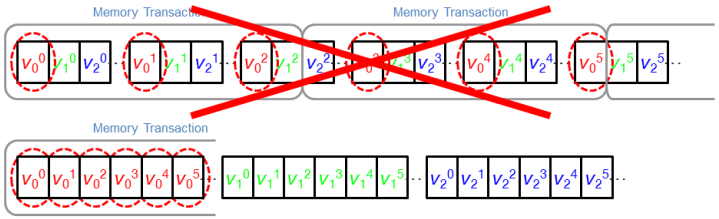
Fazit: es werden 5 x (6x4 Bytes) gelesen, obwohl nur 5x4 Bytes benötigt werden



Zugriffslokalität

- Cache-effizientes Speicherlayout

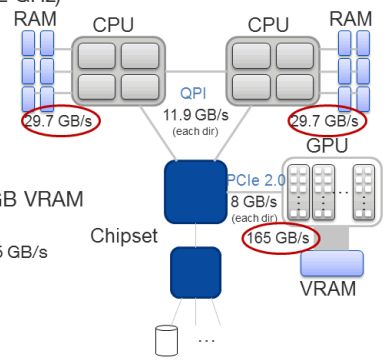
Felder mit Datenelementen, die aus mehreren Komponenten bestehen (v_0^i, v_1^i, v_2^i, …) werden häufig Komponenten-weise bearbeitet



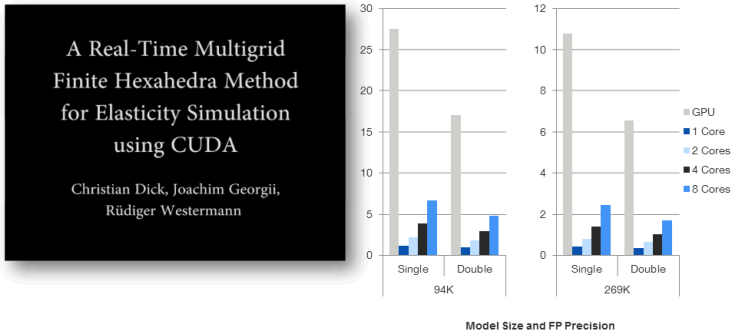
Komponenten werden in separate Speicherblöcke gruppiert, um mit möglichst wenigen Cache-Reads möglichst viele Komponenten bearbeiten zu können

Anwendungsbeispiel

- 2x Intel Xeon X5560 2.8 GHz (3.2 GHz)
  - 4 processors on one chip
- 48 GB DDR3 1333 MHz,
  - NUMA architecture
  - Theoretical memory bandwidth: 29.7 GB/s (per CPU)
  - Theoretical QPI bandwidth: 11.9 GB/s (each direction)
- NVIDIA GTX 480 (“Fermi”), 1.5 GB VRAM
  - 15 multiprocessors, 480 cores
  - Theoretical memory bandwidth: 165 GB/s



Physikalische Simulation: Zeitschritte/Sekunde



Kosten und Limitationen

Kernel	FLOPs	%	Bytes R/W		%	GPU Timings %	
			Single	Double		Single	Double
Computation of element rotations	470	2	160	300	1	0	1
Assembly of simulation level equations	10000	51	6600	13000	23	33	30
Assembly of coarse grid equations	3200	17	5100	10000	18	24	21
Gauss-Seidel relaxation	6 x 660	20	6 x 1800	6 x 3500	39	27	33
Computation of residual	2 x 620	6	2 x 1800	2 x 3500	13	11	12
Restriction of residual	2 x 210	2	2 x 580	2 x 1000	4	1	1
Interpolation of error and coarse grid corr.	2 x 39	0	2 x 200	2 x 350	1	2	1
CG solver on coarsest level	2 x 3	0	2 x 1	2 x 1	0	1	1
Total (per finite element per time step)	19000		28000	54000			

Zugriffslokalität

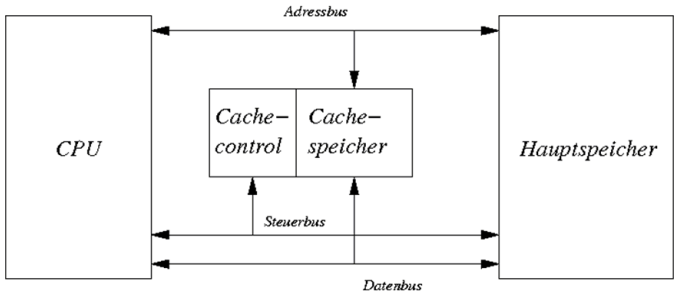
- Beobachtung: Programme greifen in einem kleinen Zeitintervall auf einen relativ kleinen Teil des Adressraums zu. Das gilt sowohl für Instruktionen als auch für Daten
- Zeitliche Lokalität
  - Wenn ein Zugriff auf eine Adresse erfolgt, wird auf diese Adresse mit großer Wahrscheinlichkeit bald wieder zugegriffen
- Räumliche Lokalität
  - Wenn ein Zugriff auf eine Adresse erfolgt, werden mit großer Wahrscheinlichkeit auch Zugriffe auf in der Nähe liegende Adressen erfolgen

Zugriffslokalität

- Beobachtung: Programme greifen in einem kleinen Zeitintervall auf einen relativ kleinen Teil des Adressraums zu. Das gilt sowohl für Instruktionen als auch für Daten
- Zeitliche Lokalität
  - Wenn ein Zugriff auf eine Adresse erfolgt, wird auf diese Adresse mit großer Wahrscheinlichkeit bald wieder zugegriffen
- Räumliche Lokalität
  - Wenn ein Zugriff auf eine Adresse erfolgt, werden mit großer Wahrscheinlichkeit auch Zugriffe auf in der Nähe liegende Adressen erfolgen

Caching

- Speicherzugriffe stellen aus Sicht des Programms bzw. des Anwenders zunächst einen Hauptspeicherzugriff dar




Ende der Bildschirmpräsentation. Zum Beenden klicken.