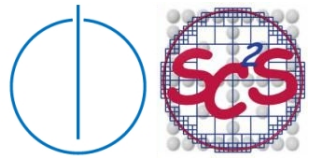
Masterpraktikum - Scientific Computing

High Performance Computing

Thomas Auckenthaler
Wolfgang Eckhardt
Prof. Dr. Michael Bader

Technische Universität München, Germany

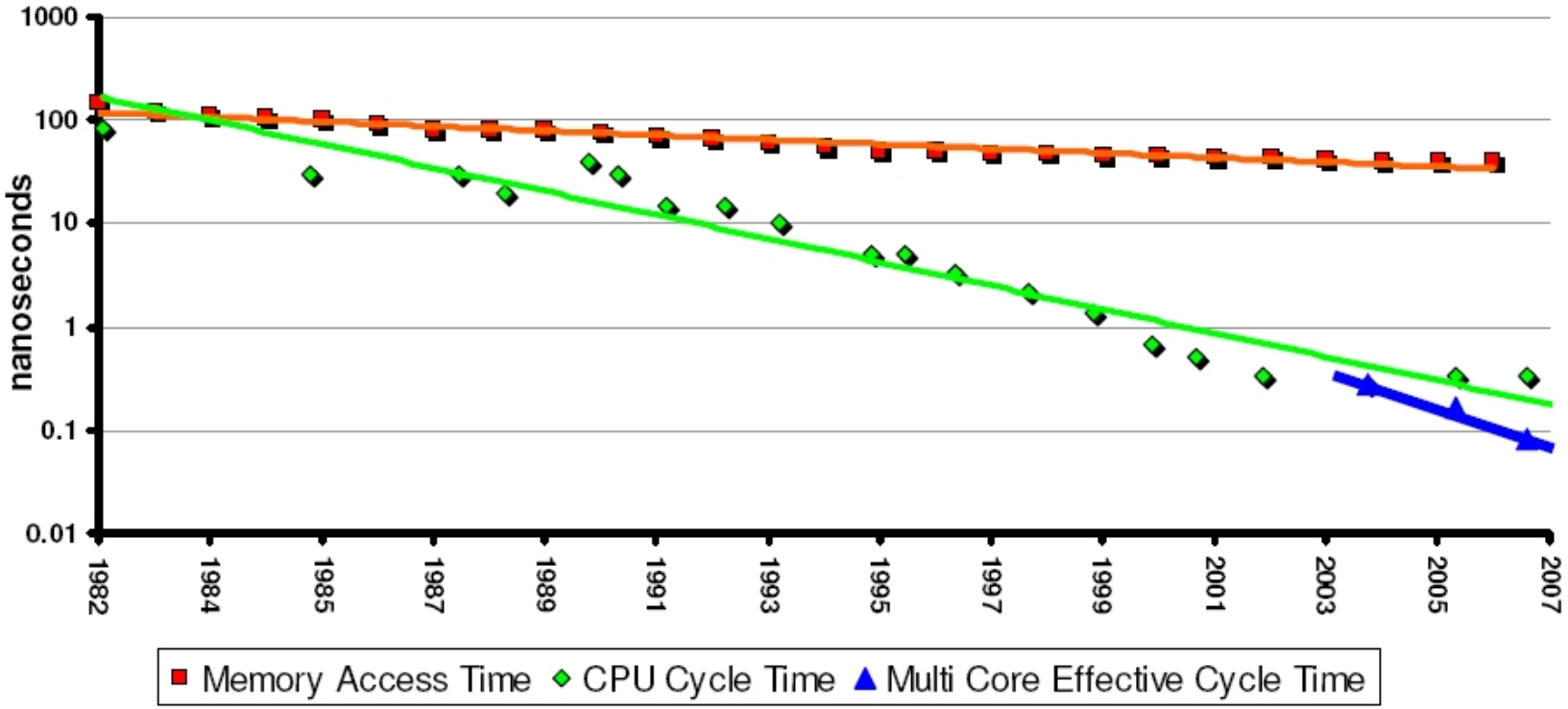


Inhalt

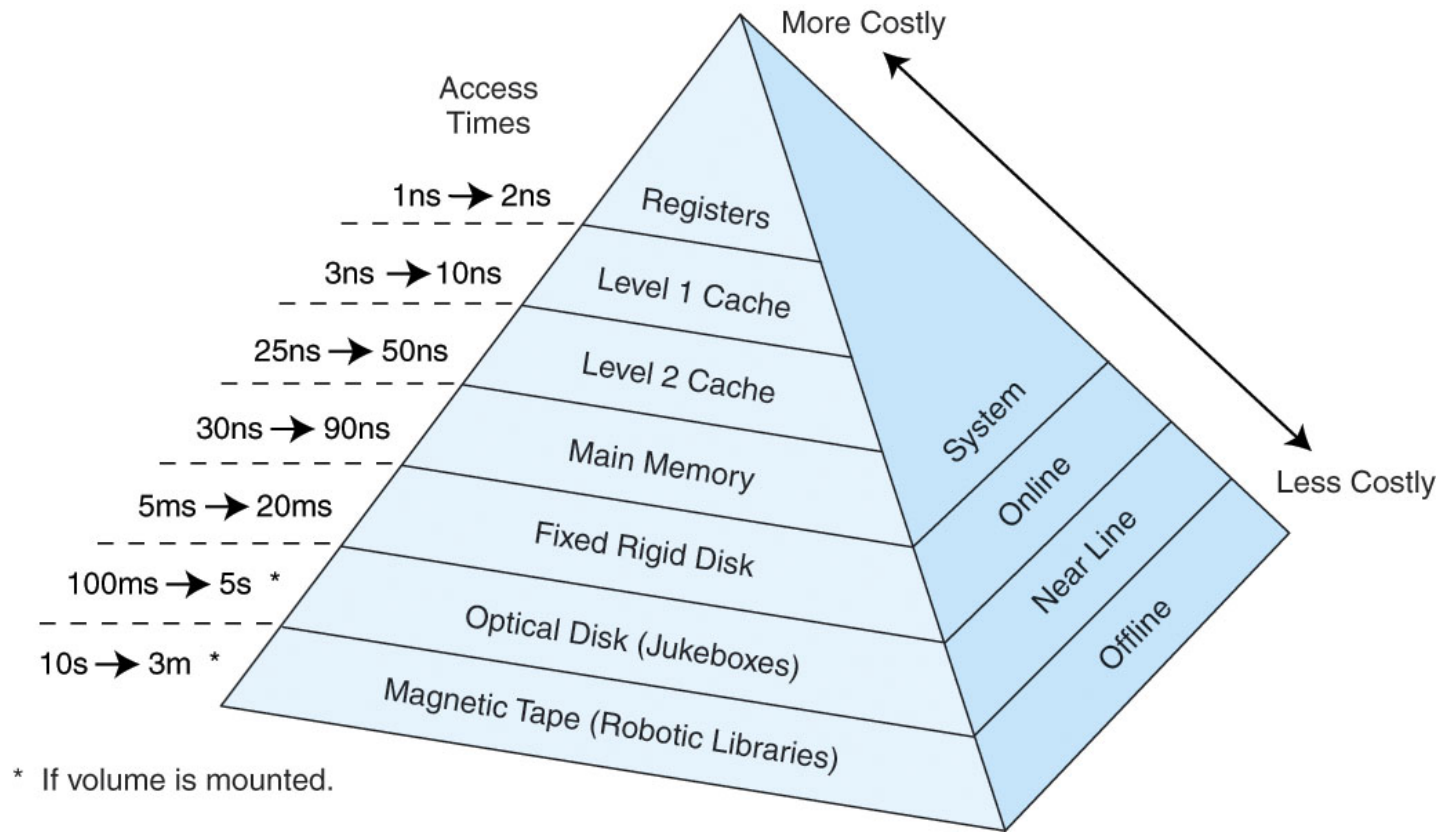
- Grundlagen zu Caches
- Cacheeffizienz durch Cache-Blocking
  - Beispiel Matrix-Matrix-Multiplikation

Caches - Motivation

Prozessor- vs. Speichergeschwindigkeit

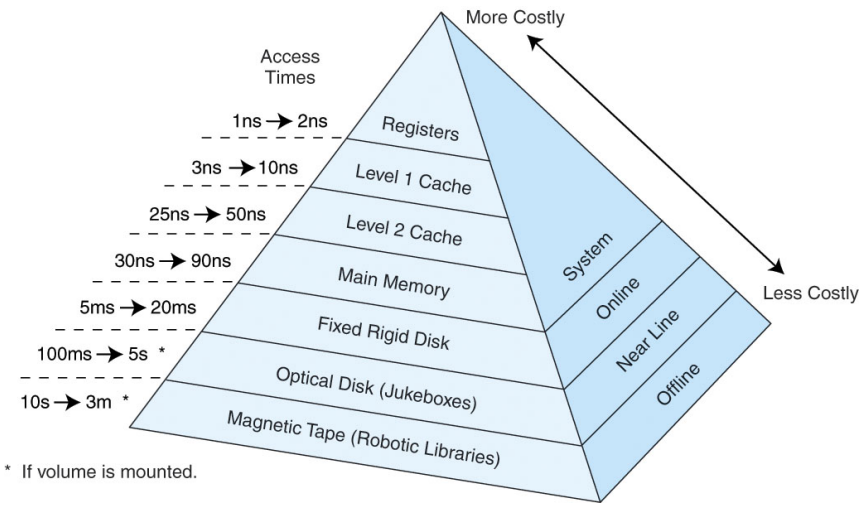


Speicherhierarchie

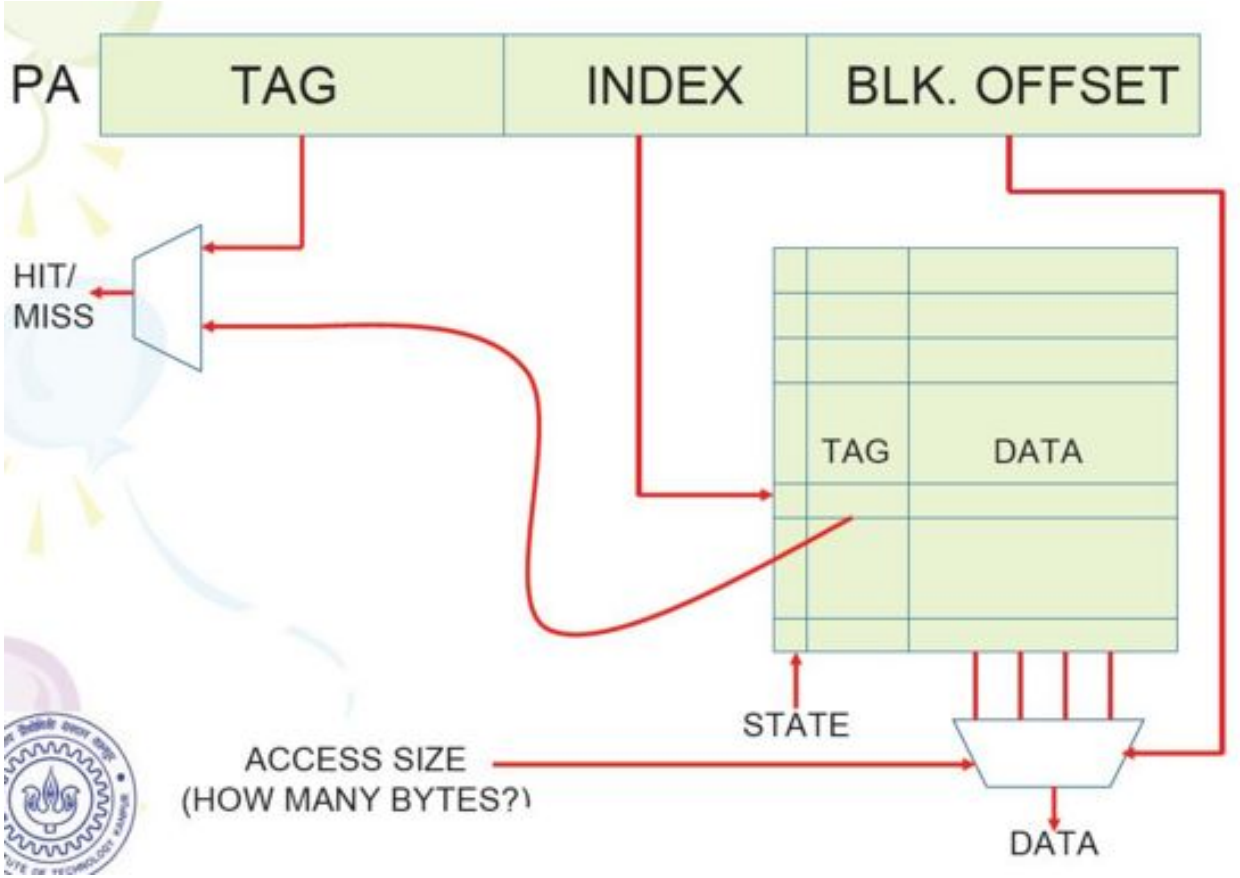


Caches

- transparenter Pufferspeicher
- Ausnutzung von Lokalitätseigenschaften:
  - **zeitliche Lokalität**: Wenn auf Adresse x zugegriffen wird, wird mit hoher Wahrscheinlichkeit kurz darauf wieder auf x zugegriffen
  - **räumlich Lokalität**: Wenn auf Adresse x zugegriffen wird, wird mit hoher Wahrscheinlichkeit auch auf Adressen in der Nähe von x zugegriffen



Caches - Aufbau



Beispiel

- 512 Cachezeilen a 64 Bytes (= 32KB)
- 32 bit Adressen (6 bit Block-Offset, 9 bit Index, 17 bit Tag)

Caches - Beispiele

- Intel Core i7
  - 32 KB L1-Datencache, 8-fach assoziativ
  - 32 KB L1-Instruktionscache
  - 256 KB L2-Cache, 8-fach assoziativ
  - 8 MB shared L3-Cache, 16-fach assoziativ
  - 64 Byte linesize
- AMD Phenom II X4
  - 64 KB L1-Datencache, 2-fach assoziativ
  - 64 KB L1-Instruktionscache
  - 512 KB L2 Cache, 16-fach assoziativ
  - 6 MB shared L3 Cache, 48-fach assoziativ
  - 64 Byte linesize

Caches - Organisation

- Mapping-Strategie (Assoziativität)
  - voll assoziativer Cache
  - direkt abbildender Cache
  - mengenassoziativer Cache
- Ersetzungsstrategie
  - First In First Out (FIFO)
  - Least Recently Used (LRU)
- Aktualisierungsstrategie
  - Write Through
  - Write Back
- Cache-Kohärenz

Cache-Misses

- **cold misses**: erster Zugriff auf Adresse (nicht vermeidbar)
- **capacity misses**: Verdrängung aufgrund der begrenzten Größe des Caches
- **conflict misses**: Verdrängung aufgrund der begrenzten Assoziativität des Caches

Cache-Blocking

- Beispiel Matrix-Matrix-Multiplikation

```
for(i = 0; i < n; i++)
 for(j = 0; j < n; j++)
   for(k = 0; k < n; k++)
     c[i,j] += a[i,k] * b[k,j];
```

Cache-Blocking

- Beispiel Matrix-Matrix-Multiplikation

```
for(i = 0; i < n; i++)
  for(j = 0; j < n; j++)
    for(k = 0; k < n; k++)
      c[i,j] += a[i,k] * b[k,j];
```

  - **Register-Blocking**: Bandbreite zum L1-Cache ist meist nicht ausreichend, um 3 Operanden pro Instruktion zu laden
  - **Cache-Blocking**: Datentransfer zum Hauptspeicher sinkt von $O(n^3)$ auf $O(\frac{n^3}{\sqrt{M}})$ (M...Cache-Größe)
  - **TLB-Blocking**: Änderung des Datenlayouts (Blocklayout), um TLB-Misses zu reduzieren