

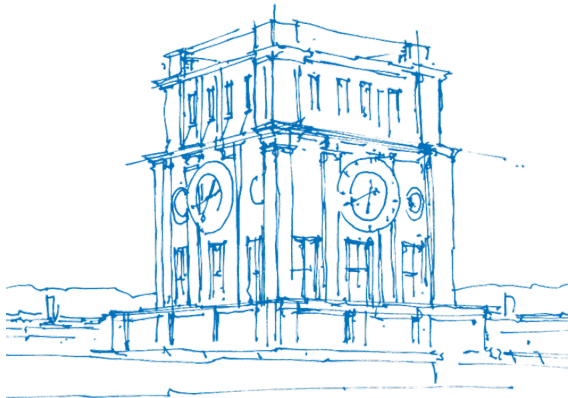
Einführung in die Rechnerarchitektur

Parallelisierung und Caches

Lukas Hertel

Lehrstuhl für Rechnerarchitektur und Parallele Systeme
Fakultät für Informatik
Technische Universität München

31. Januar 2022



TUM Uhrenturm

Wiederholung

Roofline Modell

- Applikation limitiert durch
 - Speichervolumen oder
 - Rechenleistung
- x-Achse: Arithmetische Intensität
- y-Achse: Rechenleistung
- Schätzt Top-Performance von Programmen ab

Wiederholung

Parallele Leistungs Modelle

■ Amdahl's Law

- ☐ Programme mit konstanter Problemgröße
- ☐ Sequentieller Anteil t_s und parallelisierbarer Anteil t_p
- ☐ Normalisiert durch Gesamtanteil $s = t_s / (t_s + t_p)$ (Analog für p)
- ☐ **Speedup**: $S_{Amdahl}(n) = \frac{1}{s + \frac{p}{n}}$ mit n Prozessoren
- ☐ Speedup ist beschränkt

■ Gustafson's Law

- ☐ Problemgröße steigt mit Anzahl Prozessoren
- ☐ **Speedup**: $S_{Gustafson}(n) = s + n * p$
- ☐ Speedup ist unbeschränkt

Performance von Computern

FLOP/s: Floating Point Operations per second

- Anzahl Kerne:
- Bits, die berechnet werden, pro Kern:
- Gleichzeitige Fließkommaoperationen pro Kern:
- FLOP/s:

Performance von Computern

Speicherbandbreite



Performance von Computern

Roofline Modell



Speedup

Elemente zählen

```
int count(node *head) {  
    node *cursor = head; //100ns  
    int c = 0; //1ns  
    //Solange cursor auf einen Knoten zeigt  
    while(cursor != nullptr) { //1ns  
        c++; //1ns  
        // Setzt Cursor auf den eigenen Listennachfolger  
        cursor = cursor->next; //100ns  
    }  
    return c; //1ns  
}
```



```
int x[100];  
...  
int find_max() {  
    int max_val = x[0]; //100ns  
    for(int i=0; i<100; i++) { //3ns  
        max_val = max(x[i], max_val); //17ns  
    }  
    return max_val; //1ns  
}
```