

SoundSim

Raumakustik-Simulation
Schallausbreitung und Reflexion

Frank Röder & Julius Plehn
PAPO2015

Problemstellung

- Wir haben eine $N * M * L$ Matrix auf der sich pro Zelle Frequenzteilchen befinden.
- Diese sollen nun verschiedene Frequenzbereiche mit den dazugehörigen dB Werte haben und sich bei unterschiedlicher Kollision mit Beachtung des Gesetzes der Polarität, auslöschen oder verstärken.
- Reflexionsverhalten und Ausbreitung soll simuliert werden

Algorithmus

```
int m[N][N], d[N][N];  
frequenzteilchen_id[x][y]; // Position auf dem Gitter  
int Anzahlkollisionen, Frames;
```

schicke n Teilchen von Punkt m[0][0] im "halbkreis" in eine bestimmte Richtung.

```
while(dB_vorhanden==true)  
    functionSchallwelle() // für jedes Teilchen  
        • Formung der Schallwelle  
        • Verschiebung der Teilchen, in neues Feld (nach  
Richtungsattribut)  
        • Berechnung der Kollisionen in neuem Feld  
        • visualisierenDerMatrix();
```

Zeitplan

Juli: Recherche und C-Vertiefung
Sequenzielles Programm

August: 2D -> 3D Parallelisieren & Optimieren

September: Letzte Schritte

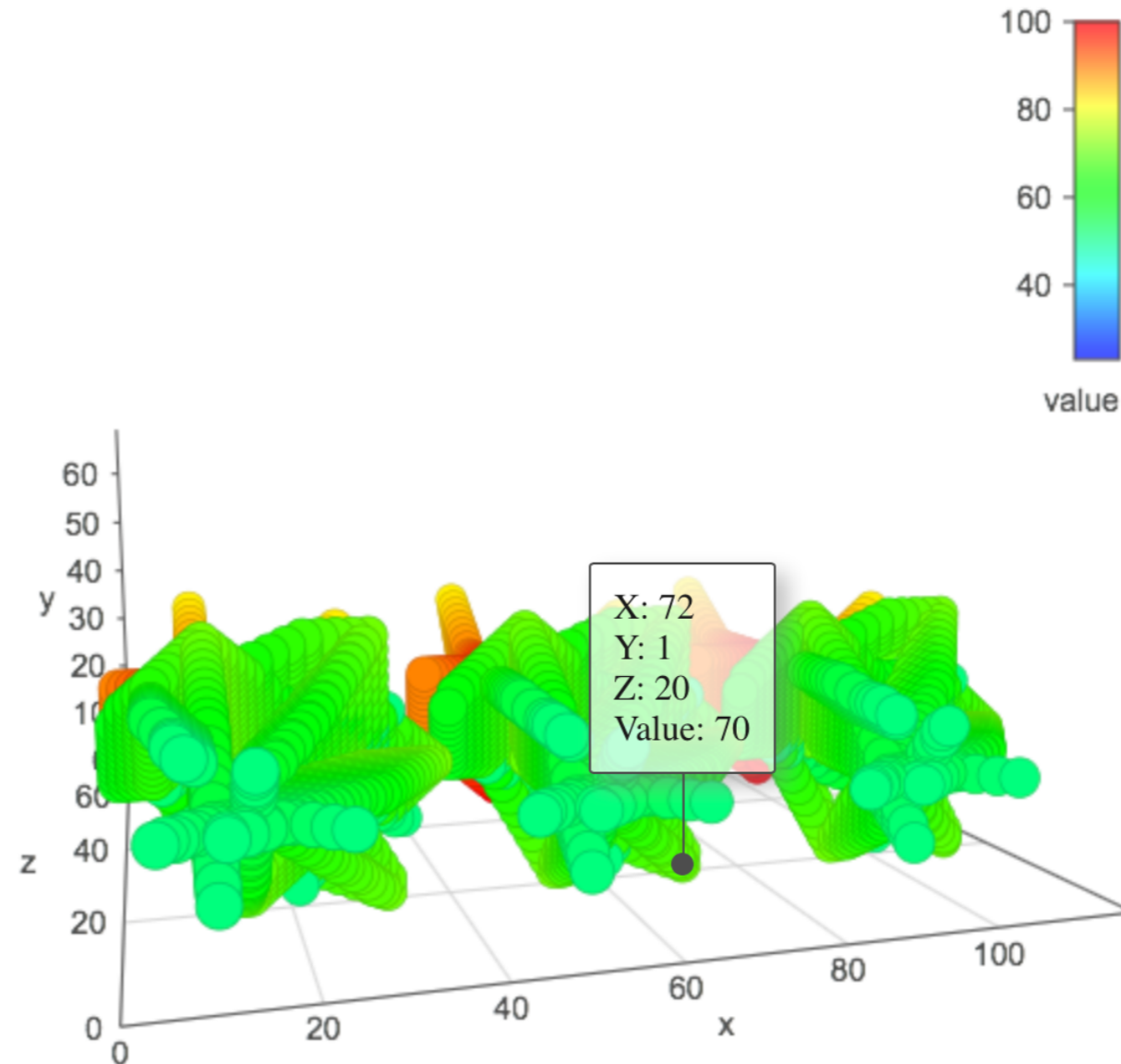
Parallelisieren

- Zerlegung der 3D-Matrix in Scheiben -> max. 2 direkte Kommunikationspartner
- jeder Prozess hat seinen Anteil am Raum
- Trennung von Verschiebung und Berechnung
- 1 Prozess zur Visualisierung

Visualisierung & Programmausgabe

time: 19

mit .json

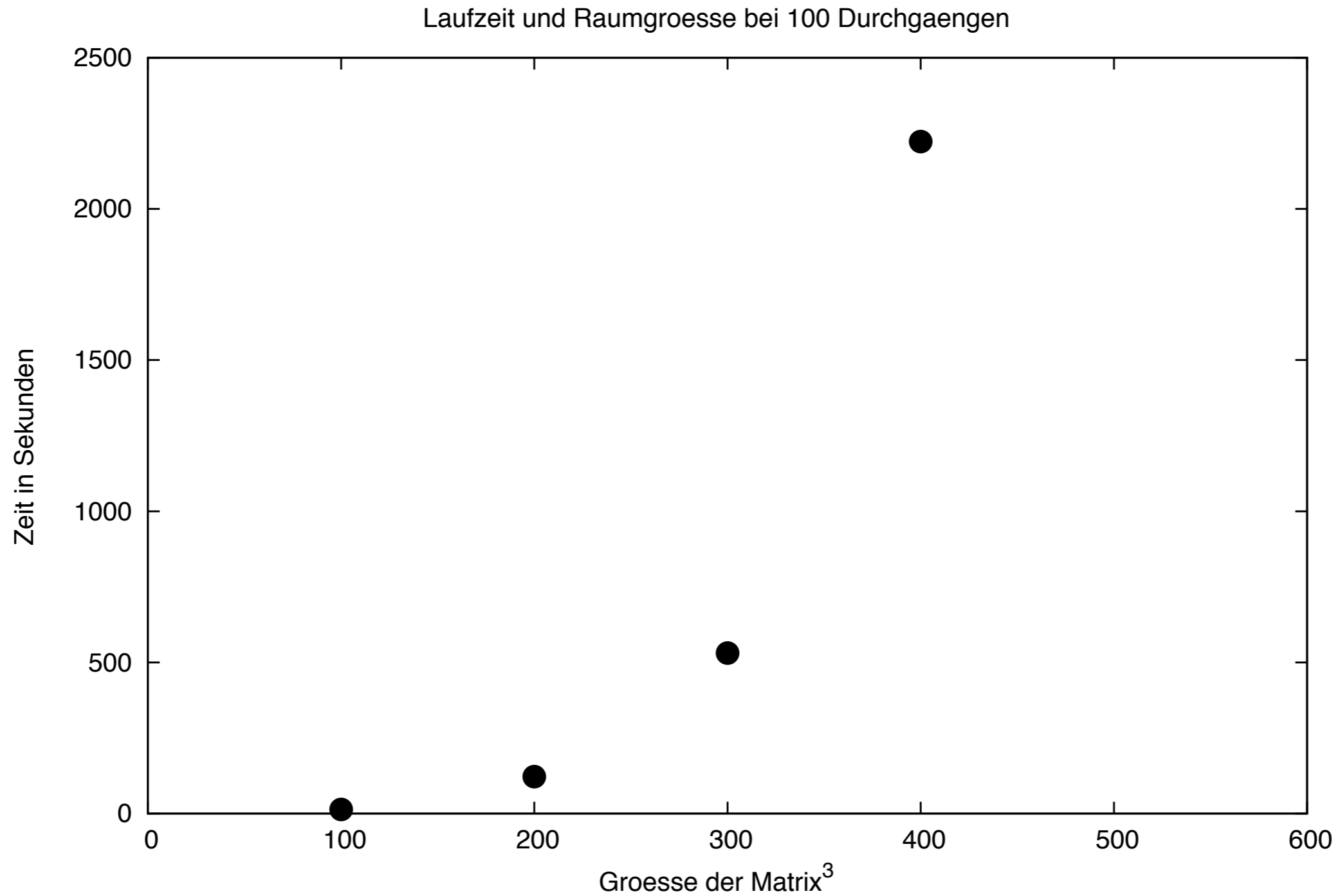


Prev

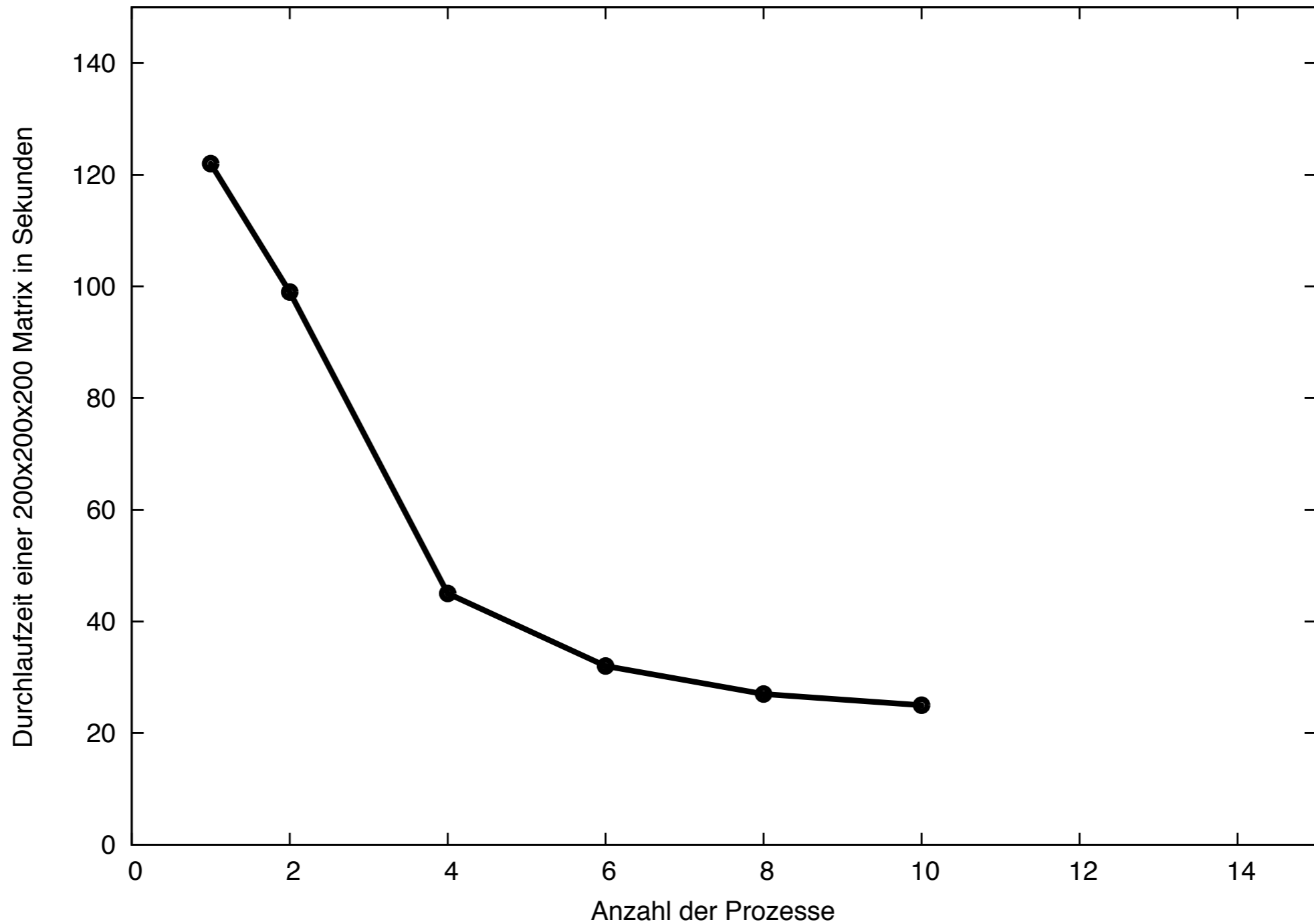
Play

Next

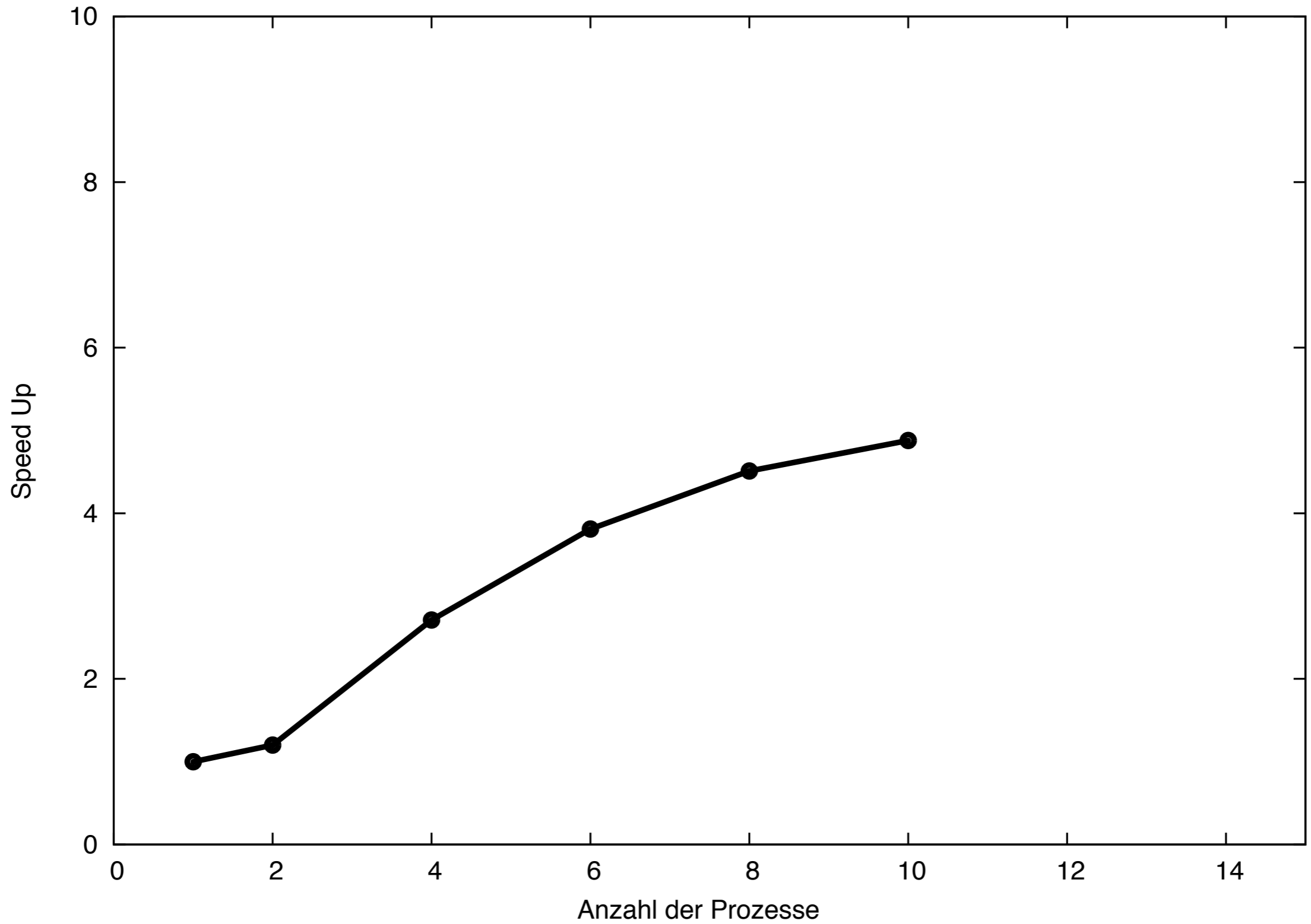
Laufzeitmessungen Sequenziell



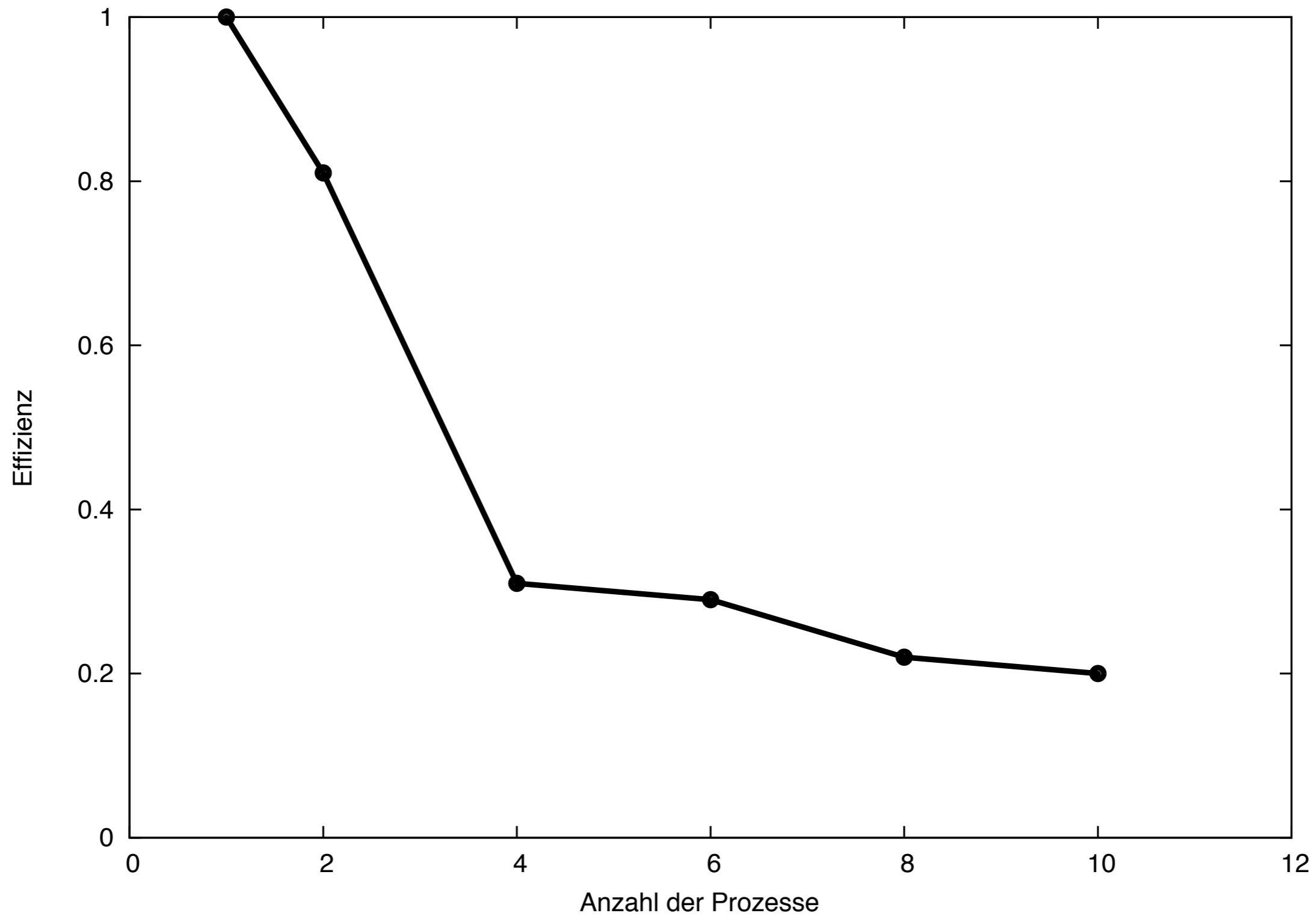
Laufzeitmessung Parallel



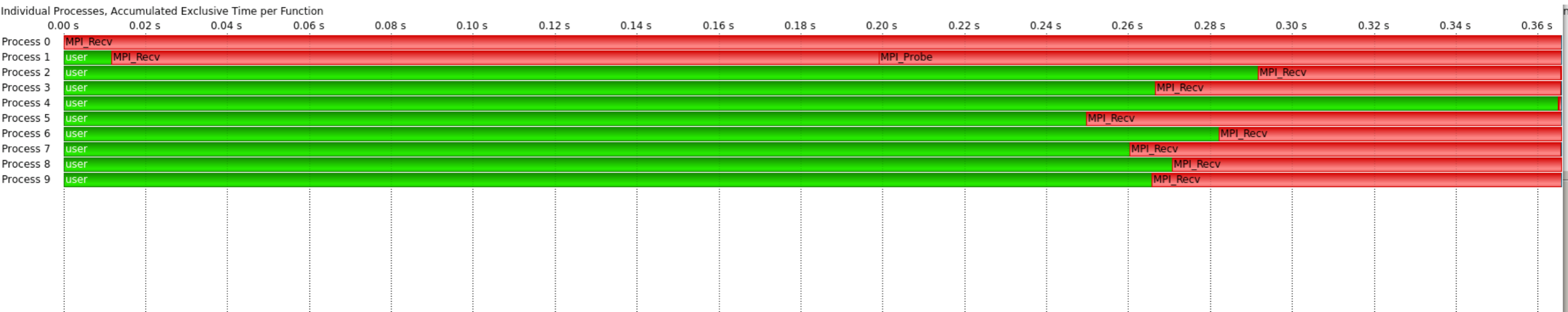
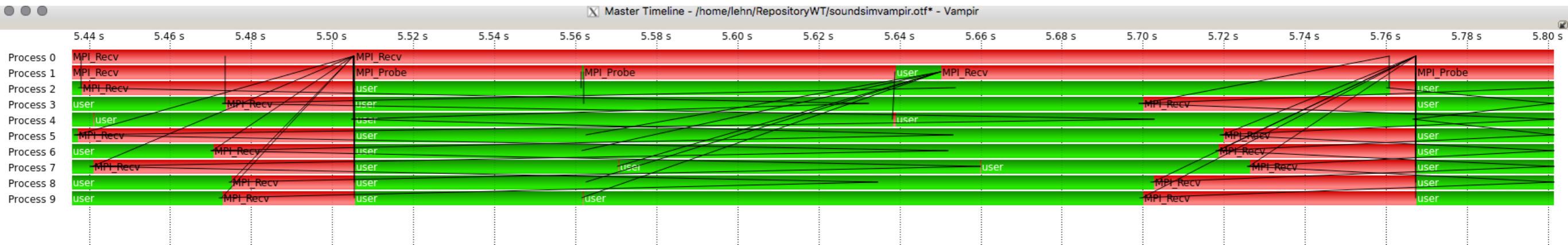
Speedup



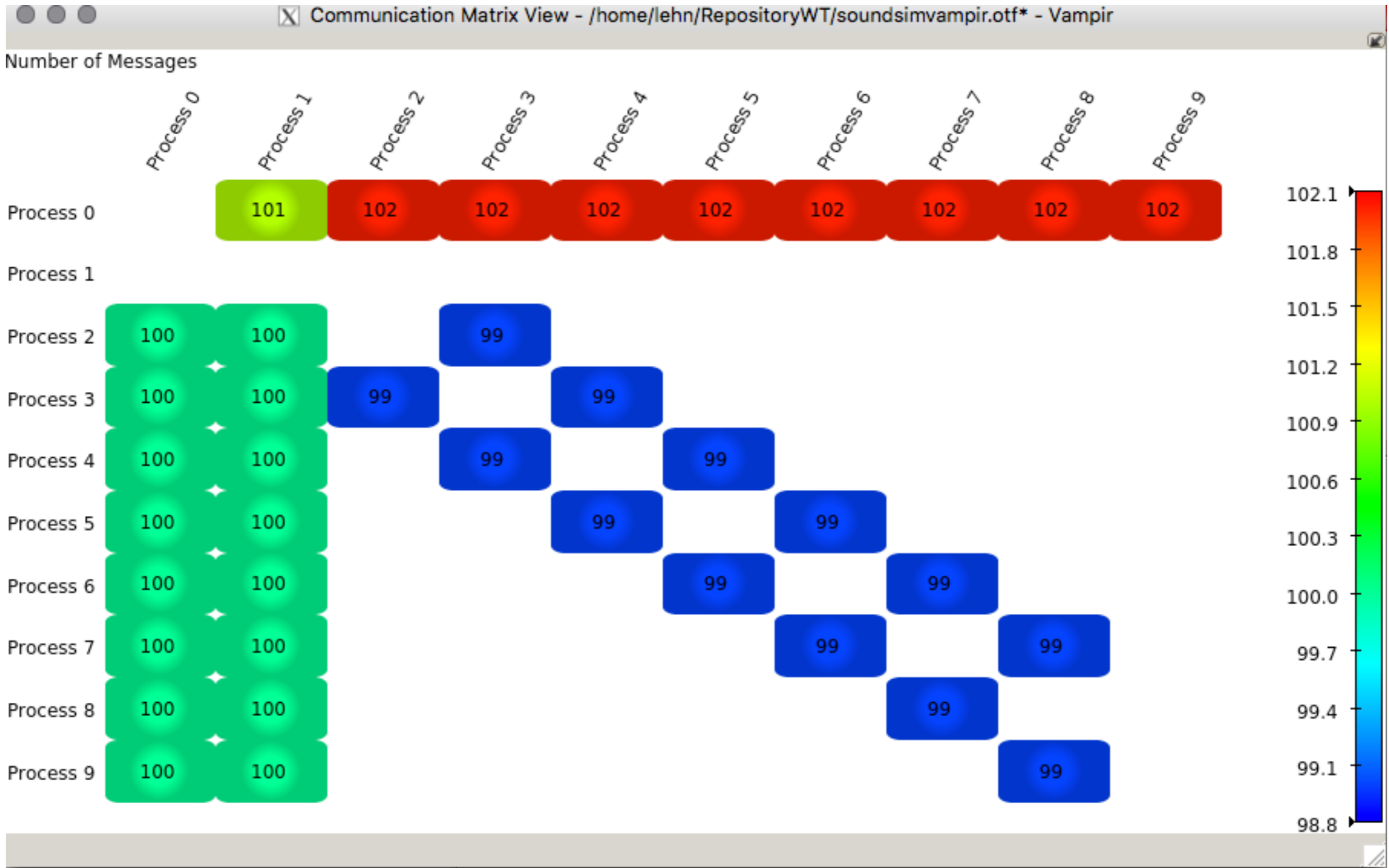
Effizienz



Vampir Timeline



Vampir Kommunikation



Skalierbarkeit

- Auflösungsbedingte Anzahl an Prozessen/Raum
- Vorsicht vor gebündeltem Rechenaufwand