

Bei dieser Messungsreihe sollte die Laufzeit bei unterschiedlicher Anzahl Threads untersucht werden. Hierfür wurden Messungen mit folgenden Parametern vorgenommen:  
 $x \cdot 2 \cdot 512 \cdot 2 \cdot 2 \cdot 1000$ , wobei  $x$  die Anzahl der Threads ist. Aus den Messungen ist folgendes Diagramm entstanden.

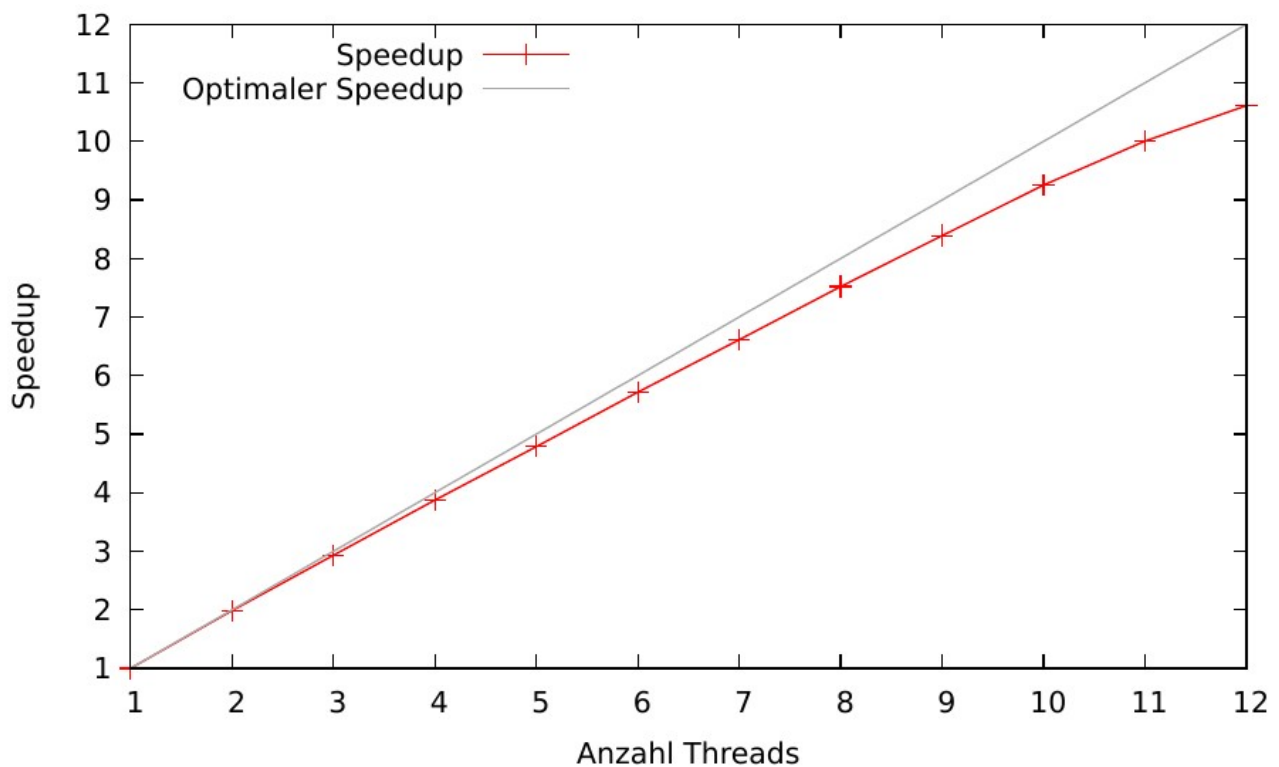


Abbildung 1: Speedup-Graph

Zusätzlich zu der Visualisierung der Messergebnisse (in rot) ist eine Ideallinie (in grau) eingetragen, die den optimalen Speedup darstellt. Die Ergebnisse bleiben für jede Anzahl Threads hinter dem Optimum zurück. Ebenfalls fällt beim Vergleich mit dem ursprünglichen sequentiellen Programm auf, dass das parallelisierte Programm mit einem Thread ca 10% langsamer ist. Der Grund hierfür ist, dass die Nutzung von pthreads einen gewissen Overhead mit sich führt (Erzeugung der Threads, Passieren von Barriers, Lock und Unlock von Mutexes), der im sequentiellen Programm selbstverständlich nicht vorhanden ist. Dieser Overhead ist auch der Grund dafür, dass das mit pthreads parallelisierte Programm konstant hinter dem Optimum bleibt. Aufgrund der Barriers und Mutexes, die für die Synchronisierung des Programms erforderlich sind, können die Threads nicht ungehindert durchlaufen.

Anmerkung zur Bearbeitungszeit: Ca 5 Stunden, wovon 4 für die Fehlersuche waren.

Alle Messungen auf west3

Sequentiell:

527,754s

527,472s

527,362s

12 2 512 2 2 1000:

53,949s

54,661s

53,534s

11 2 512 2 2 1000:

56,799s

57,123s

58,031s

10 2 512 2 2 1000:

62,190s

61,696s

62,030s

9 2 512 2 2 1000:

68,264s

68,478s

68,284s

8 2 512 2 2 1000:

76,393s

76,282s

76,132s

7 2 512 2 2 1000:

86,765s

86,568s

86,856s

6 2 512 2 2 1000:

100,429s

100,402s

100,105s

5 2 512 2 2 1000:

119,997s

119,879s

119,547s

4 2 512 2 2 1000:

147,859s

148,584s  
147,824s

3 2 512 2 2 1000:  
195,370s  
196,600s  
195,165s

2 2 512 2 2 1000:  
288,866s  
288,818s  
289,084s

1 2 512 2 2 1000:  
572,670s  
574,011s  
573,663s