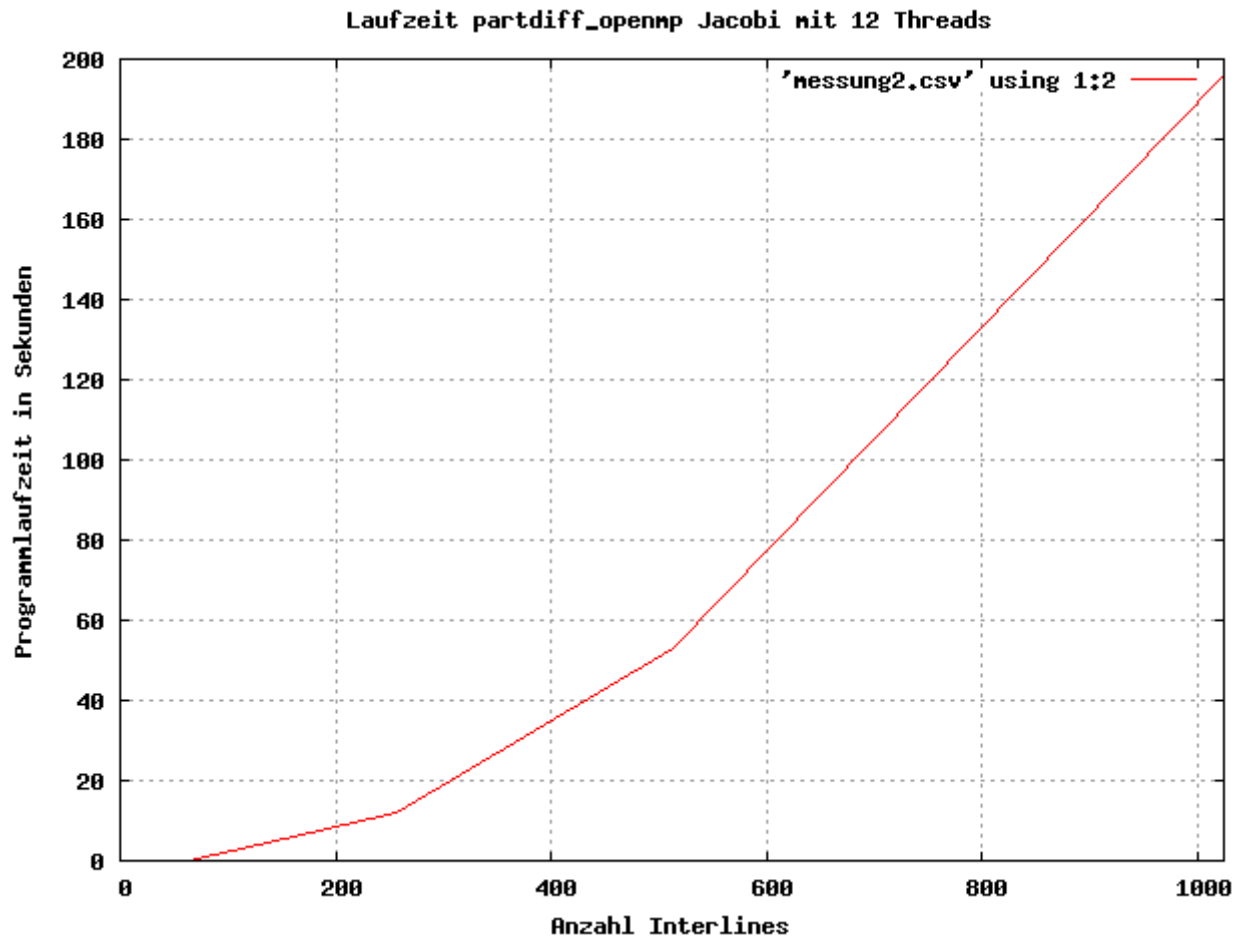


Messung 1:

Es ist zu sehen, dass die Programmlaufzeit mit steigender Threadanzahl sinkt. Dabei verlaufen Threadanzahl und Programmlaufzeit fast gegenproportional. Das bedeutet, dass das Programm beispielsweise bei doppelter Threadanzahl nur die halbe Laufzeit aufzuweist. Die Programmlaufzeit beträgt dann jedoch nicht nur die Hälfte, sondern ein wenig mehr. Der Anteil des Programms, der bei Hinzunahme weiterer Threads nicht beschleunigt wird, stellt sich wohl aus den nicht parallelisierten Codeabschnitten und dem Overhead der Parallelisierung zusammen (Aufteilung der Aufgaben, Zusammenführung der Ergebnisse).



Messung 2:

Hier ist erkennbar, dass die Programmlaufzeiten für weniger als 100 Interlines so gering sind, dass sie im Diagramm gerundet 0 betragen. In diesen Fällen sind die Berechnungsmatrizen des Programms so gering, dass auch die Programmlaufzeit sehr klein ausfällt.

Für größere Werte für die Interlines erkennt man einen stets stärker werdenden Anstieg der Programmlaufzeit. Die Laufzeit scheint also quadratisch mit der Anzahl der Interlines zu wachsen. Das entspricht außerdem der Größe der Matrix. Es sieht so aus, dass unabhängig von der Parallelisierung mit Threads ein quadratischer Anstieg der Programmlaufzeit besteht.