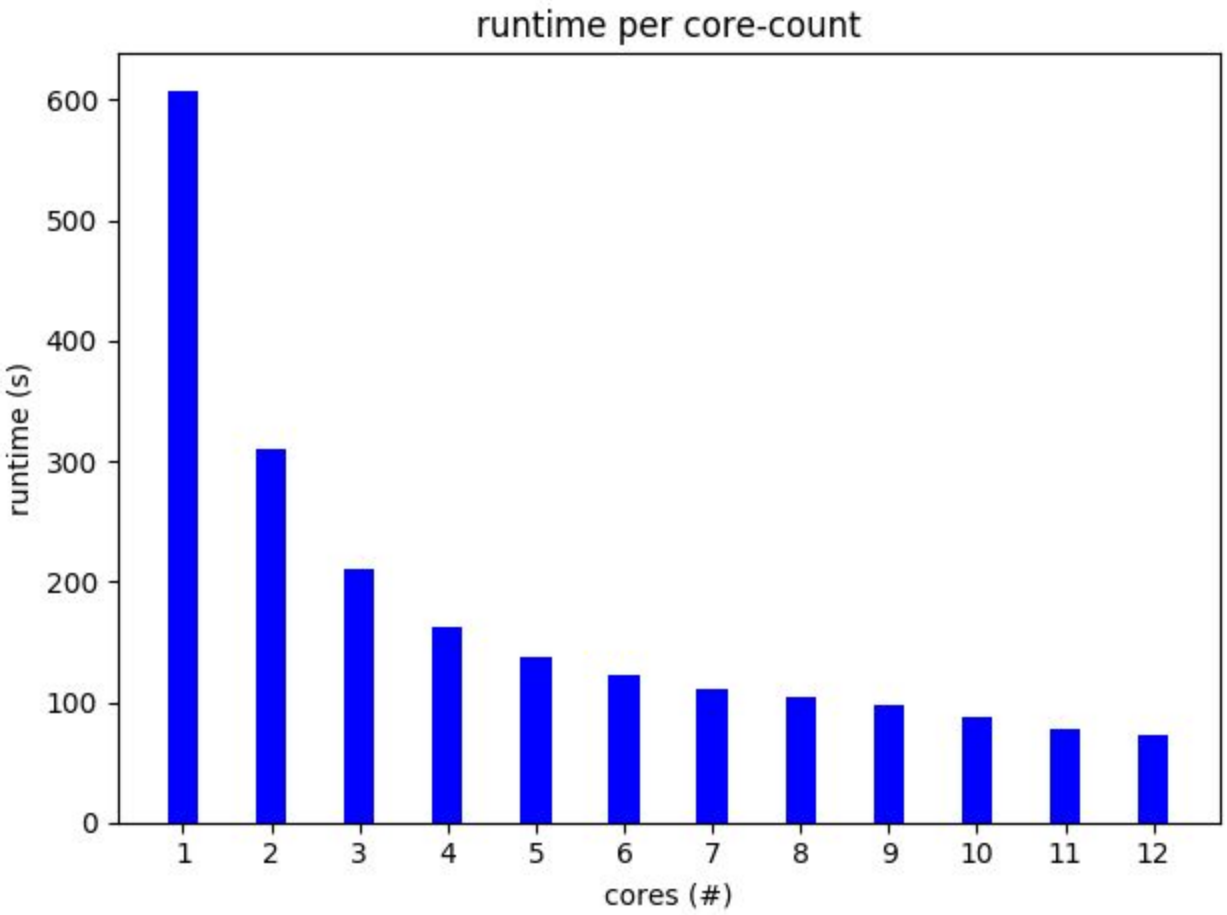
Die pthread-Implementation hat ca. 2h gedauert. Davon war 1h für die Fehlersuche notwendig.

Leistungsanalyse

Für die Messungen führen wir das Programm mit folgenden Parametern aus:
`./partdiff i 2 512 2 2 1152 mit 1<=i<=12`

Laufzeit des sequentiellen Programms:
559.0791s

Laufzeit des parallelisierten Programms mit 12 Threads:
73.0296s (7.7x)



Bei geringer Anzahl von threads ist eine gute Skalierung erkennbar. Z.B. ist die Laufzeit mit 2 threads (310.301359s) nur geringfügig größer als die halbe Laufzeit mit nur einem Thread, welche eine ideale Skalierung bedeuten würde (608.048553s / 2 ~= 304.024277s). Allerdings zeigt sich, dass mit zunehmender thread-Anzahl der Abstand zwischen idealer und tatsächlicher Laufzeit wächst und die Berechnung schlechter skaliert. Dies ist teilweise auf den zusätzlichen Aufwand zurückzuführen, der für die Erstellung der threads benötigt wird. Ein weiterer Faktor ist, dass in unserer Implementation am Ende jeder Iteration auf alle threads gewartet wird, um den aktuellen Wert von maxresiduum festzulegen. Dabei kann es vorkommen, dass kürzer laufende threads auf länger laufende warten müssen.

Alle Messwerte liege in graph_05.py vor.