

Für die einfache Parallelisierung haben wir nur einige Minuten gebraucht. Wir haben uns entschieden die äußere for-Schleife mithilfe der openmp-Compiler-Direktive

"#pragma omp parallel for"

zu parallelisieren. Für die Messungen führen wir das Programm mit folgenden Parametern aus:

```
./partdiff 1 2 512 2 2 1024
```

Laufzeit des sequentiellen Programms:

559.0791s

Laufzeit des parallelisierten Programms mit 12 Threads:

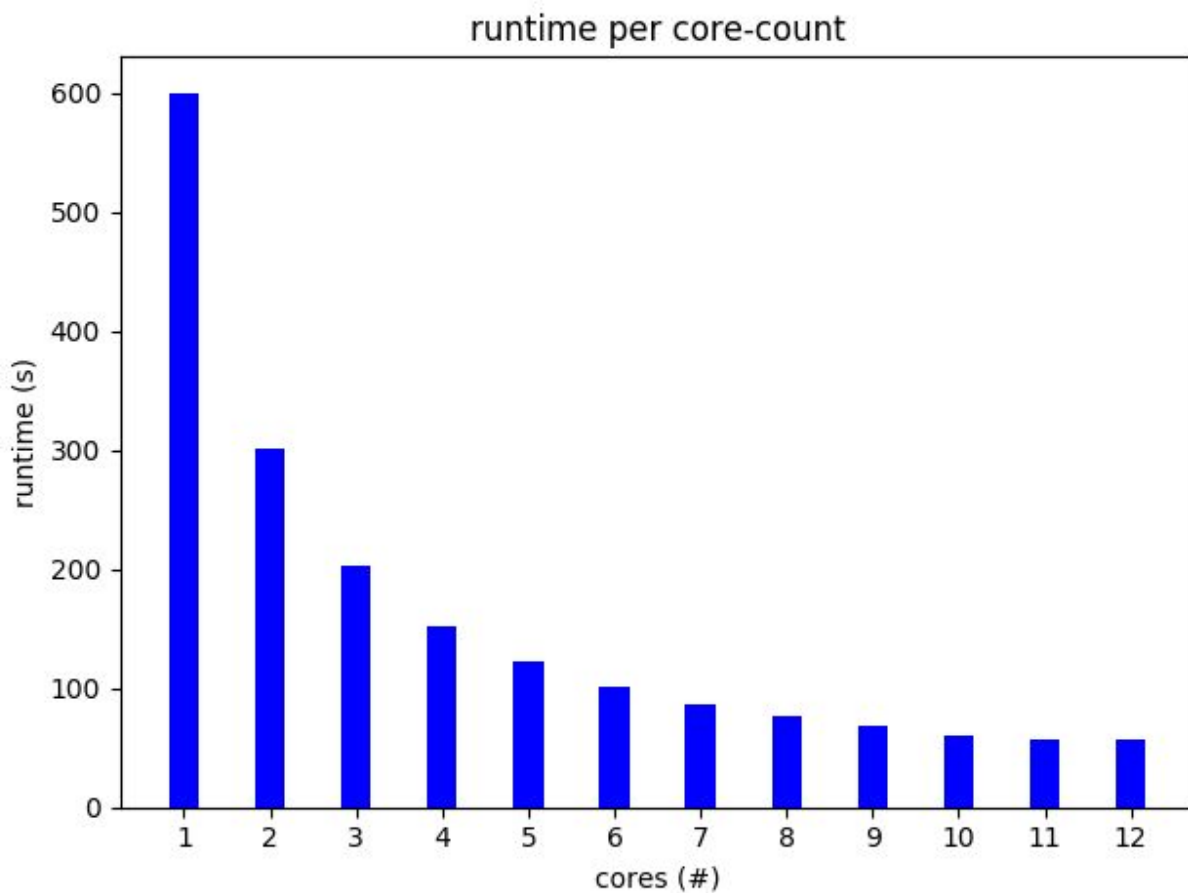
47.9273s (11.67x)

Leistungsanalyse

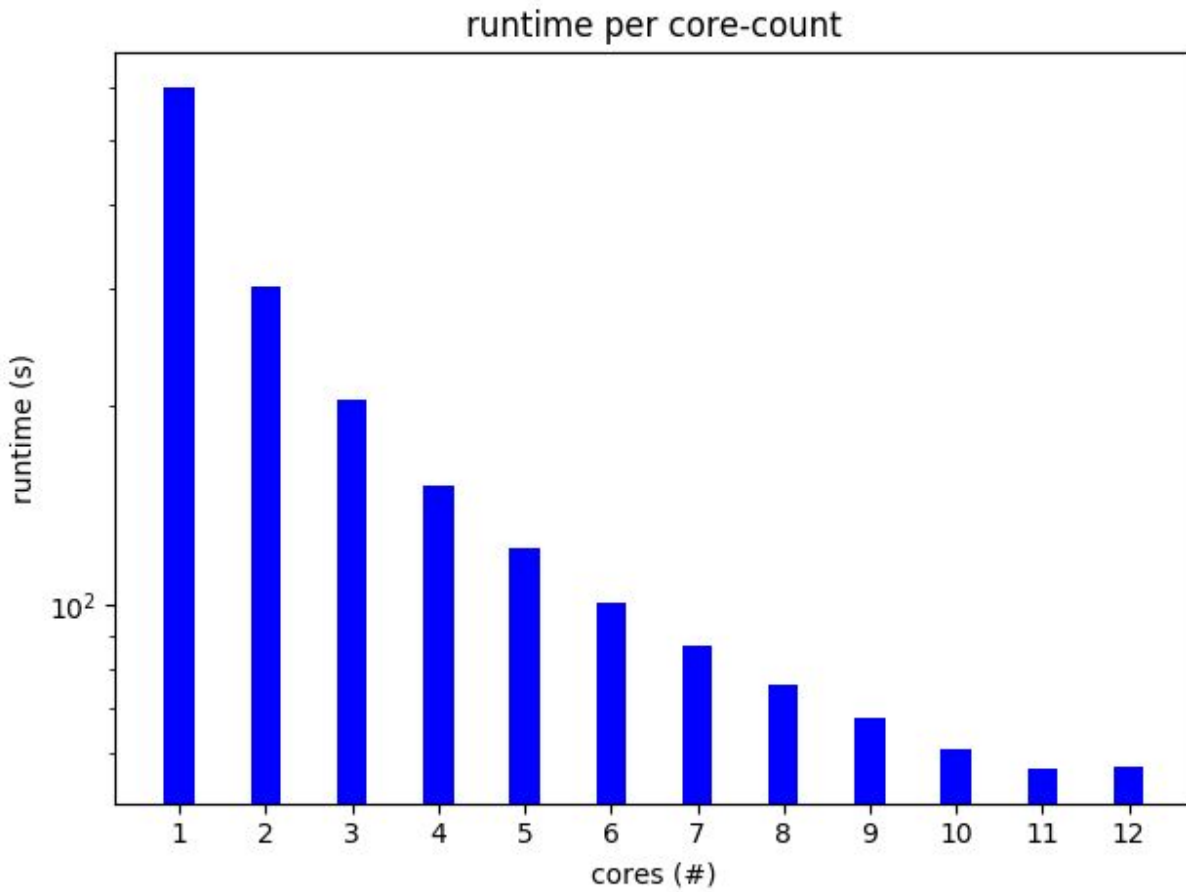
Messung 1

Für diese Messung verwenden wir die Parameter:

```
./partdiff 1 2 512 2 2 1152
```



Um die Skalierung besser bewerten zu können generieren wir zusätzlich ein Diagramm mit logarithmischer Skala:

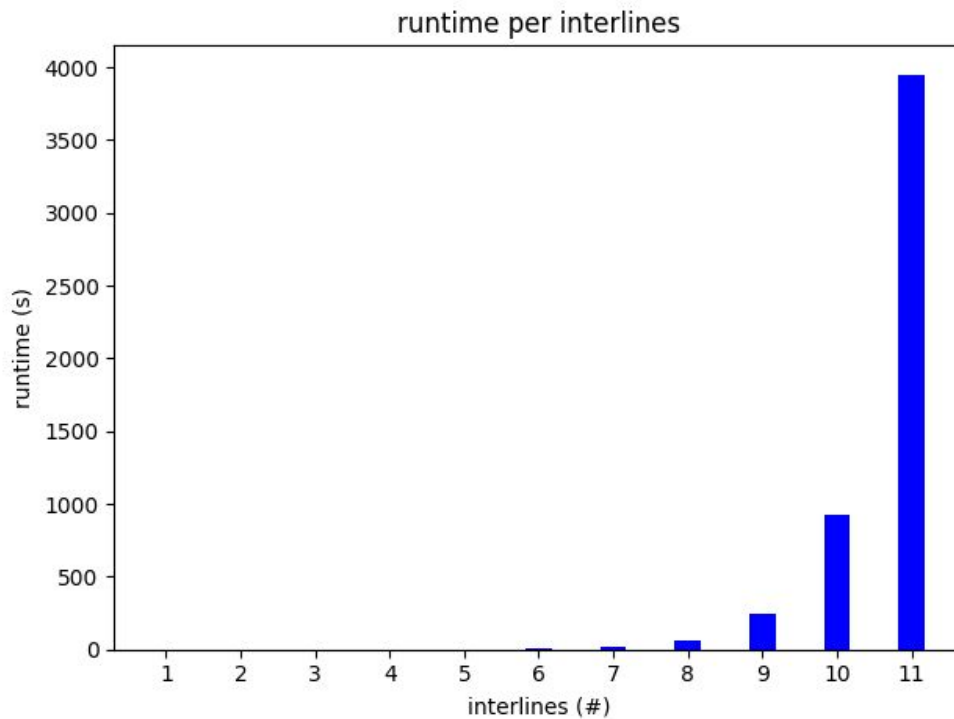


Bei perfekter Skalierung würde im Graph eine Linie zu sehen sein. Dies ist zwar nicht exakt der Fall, allerdings kommen die Ergebnisse dem sehr nahe.

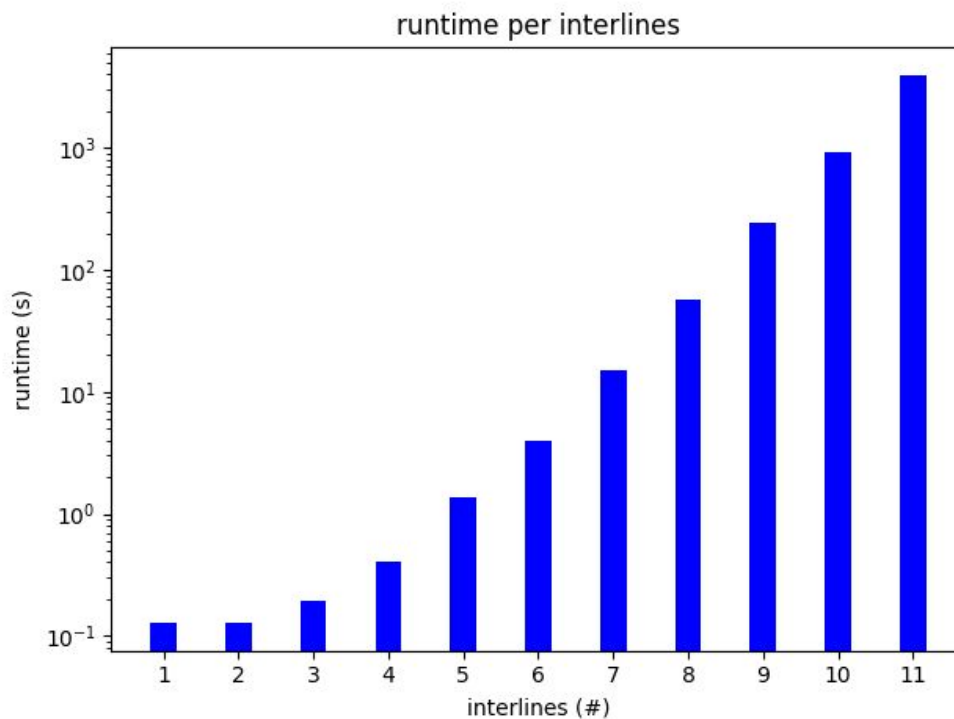
Messung 2

Für die zweite Messung verwenden wir die Parameter:

```
./partdiff 1 2 2^i 2 2 20480 für 0 <= i <= 10
```



Auch hier bietet sich eine logarithmische Skala an:



Man erkennt, dass der Rechenaufwand sich nicht wie die Eingabe verdoppelt, sondern exponentiell wächst.