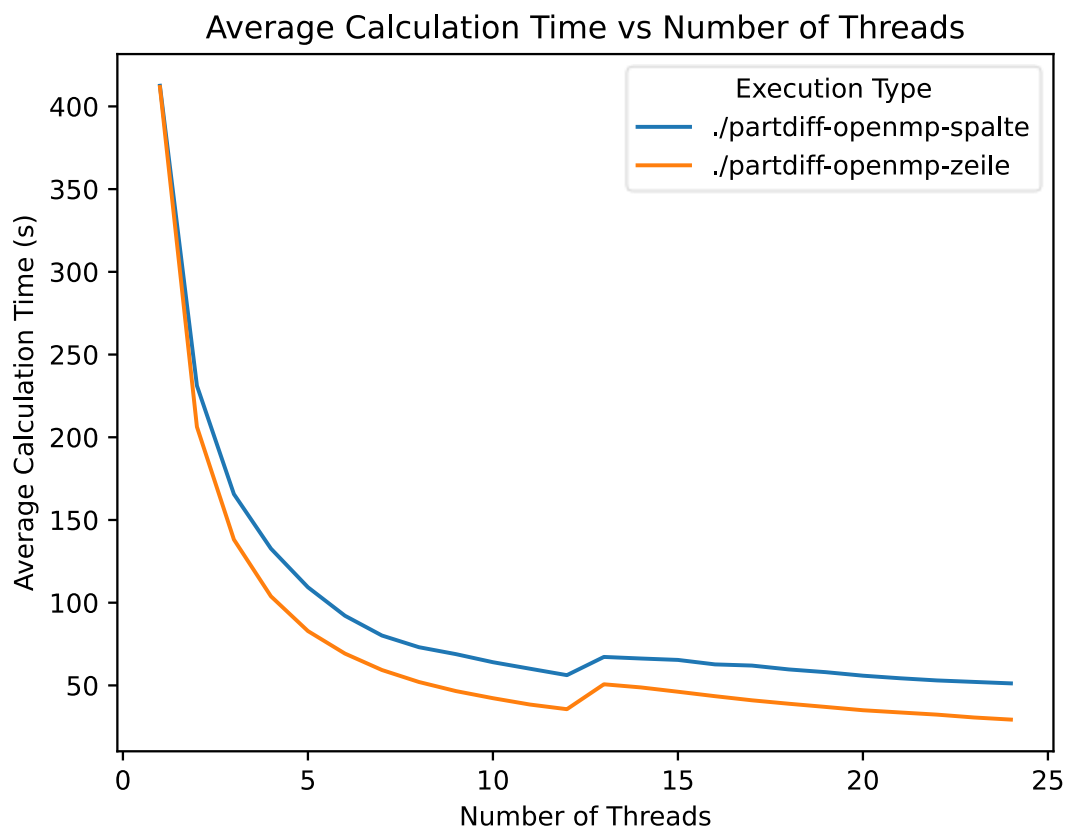


Aufgabe 3 Messung 1

Parameter:

Alle Messungen wurden auf dem Knoten West1 ausgeführt

`./partdiff-openmp-[zeile | spalte] <Threadcount> 2 512 2 2 1000`



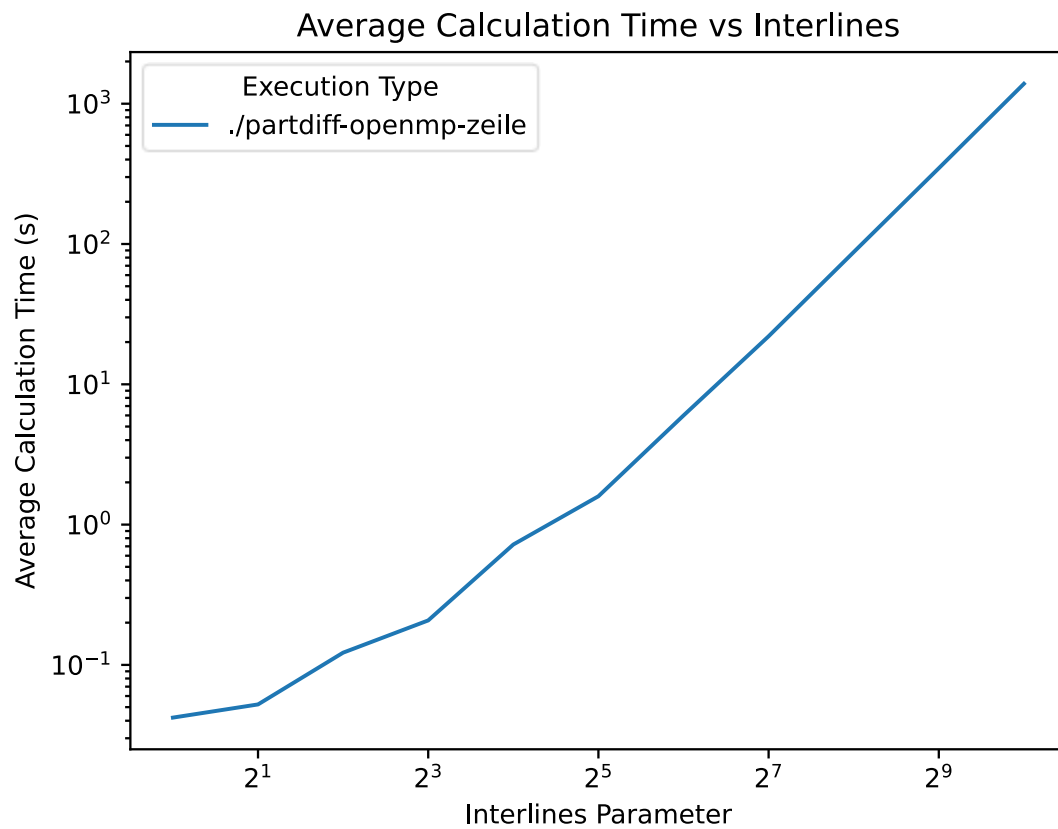
Mehr Threads verringern die Laufzeit, grundsätzlich bis die 12 physikalischen Kerne ausgelastet sind. Bei weiteren logischen Kernen wird die Laufzeitverbesserung geringer. Jedoch sind 24 Kerne bei unseren Messungen schneller als 12 Kerne.

Aufgabe 3 Messung 2

Parameter:

Alle Messungen wurden auf dem Knoten West1 ausgeführt

`./partdiff-openmp-zeile 12 2 <interlines> 2 2 10000`



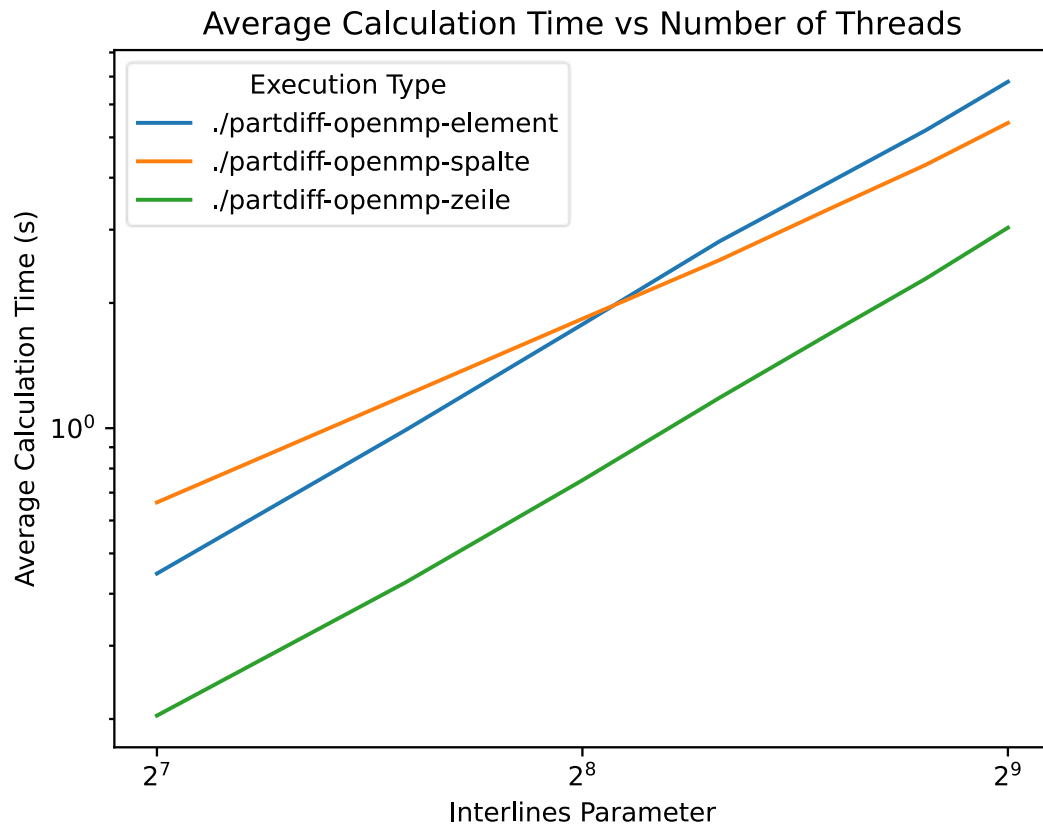
Grundsätzlich wächst die Rechenzeit mit dem Quadrat des Interlines Parameters, was den Erwartungen entspricht. Bei kleineren Werten von Interlines ist im Graphen ein Overhead erkennbar.

Aufgabe 2a

Parameter:

Alle Messungen wurden auf dem Knoten West1 ausgeführt

`./partdiff-openmp--[zeile | spalte | element] 24 2 <interlines> 2 2 100`



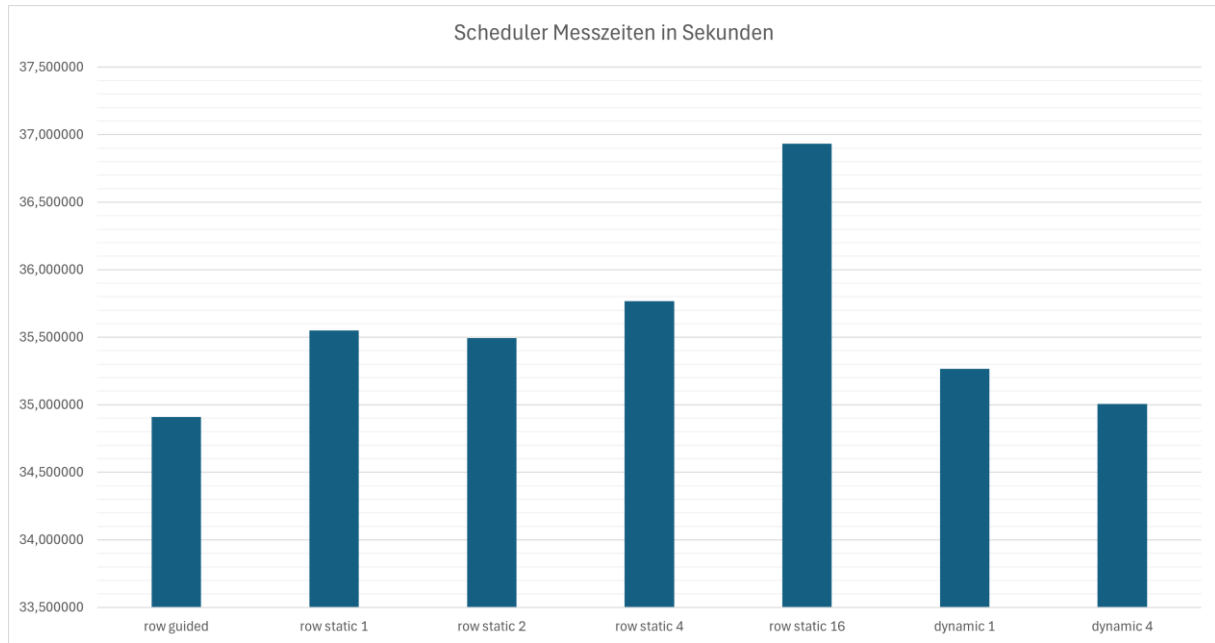
Die Zeilenweise Implementierung ist bei unseren Messungen die beste. Die Elementenweise und Spaltenweise Implementierung haben ähnliche Ergebnisse. Dies kann an einer teilweise fehlerhaften Implementierung der Spaltenweisen-Methode liegen.

Aufgabe 2b

Parameter:

```
./partdiff-openmp-zeile 12 2 128 2 2 1000
```

Alle Messungen wurden auf dem Knoten West3 ausgeführt, Sourcecode von Partdiff wurde der Zeilen-Messung für die for Schleife entsprechend angepasst



Guided ist der beste Scheduling Algorithmus, da er sich dynamisch auf Basis der geleisteten Arbeit anpasst.

Dynamic ist der nächstbeste Algorithmus, der mit zunehmender Blockgröße von 1 auf 4 effizienter wird.

Static ist hier der schlechteste Algorithmus, der bei zunehmender Blockgröße von 1 bis 16 schlechter wird.