

Vorgehensweise

Ich verwende ein ähnliches Prinzip wie bei der ersten Threading Aufgabe: Aufteilen der Range in mehrere Teile, wobei der letzte Teil größer als die anderen ist. Damit es kein Einheitsbrei wird habe ich dieses mal innerhalb des zuletzt laufenden Threads die fehlenden zahlen summiert.

Aufwand

Ich weiß nicht was sie hier erwarten, aber an sich wurde dies alles innerhalb von ~4h geschrieben.

Resultate, Beobachtungen und Interpretation

Es funktioniert! *böses Lachen*

Ich verstehe nicht wieso die Laufzeit bei bis zu 4 zusätzlichen Threads bei gerader Threadanzahl größer ist (0.0005 Sekunden bei Ungeraden, 0.001 Sekunden bei geraden bei Summe von 1 bis 10).

Was ich hingegen gut verstehe ist wieso ab dem 9. zusätzlichen Thread die Laufzeit konstant doppelt so hoch ist wie vom 5. bis zum 8. (0.001 Sekunden bei 5-8, 0.002 Sekunden bei 9 und 10) bis zum 8. Thread ist die CPU mittels Hyperthreading (Ich habe in diesem Fall zusätzliche 4 Simulierte Kerne! Also in Summe 8) in der Lage sie komplett parallel laufen lassen zu können. Der 9. Thread muss dann jedoch warten bis er an die Reihe kommt.

Schwierigkeiten

Ich konnte nicht herausfinden wie ich die Performance optimieren kann.

Conclusio

Much random, very ezy, so done.

