

# Leistungsanalyse

Klaus-Johan Ziegert

Tobias Knoppler

November 14, 2015

## 1 Partdiff-seq Data

Hier wurde die Funktion Partdiff-seq mit 512 Interlines und 65 Iterations aufgerufen.

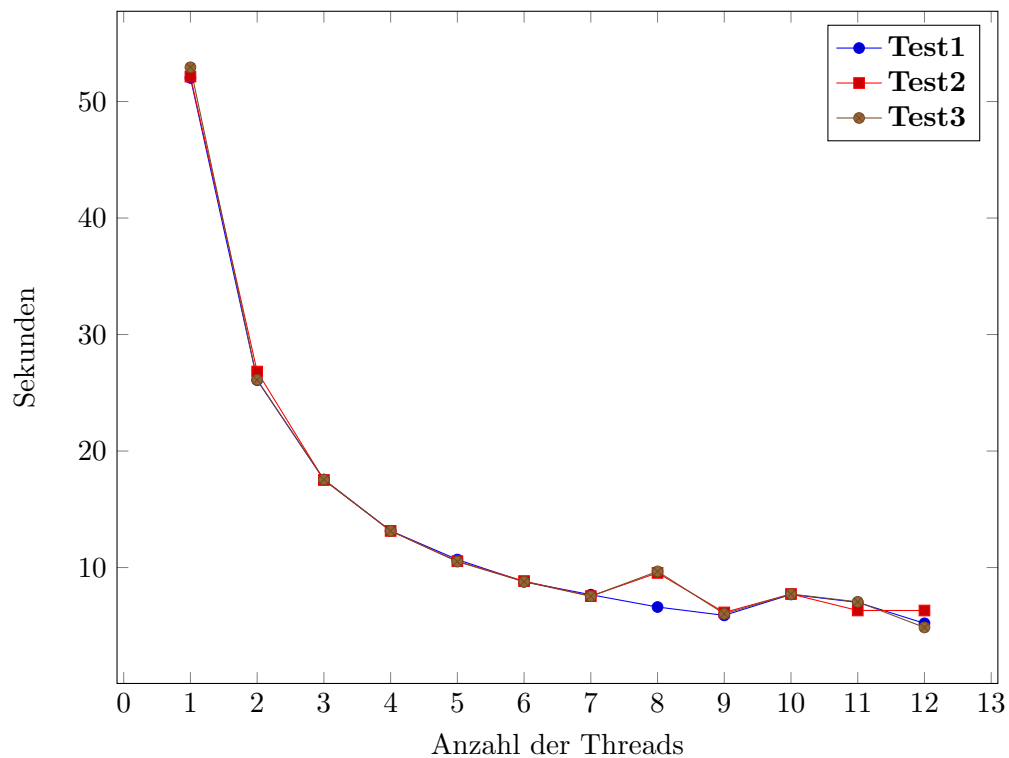
Anzahl der Threads \ Test	1	2	3
1	52.018536	51.960722	51.954834

## 2 Partdiff-posix Data

Hier wurde die Funktion Partdiff-posix mit 512 Interlines und 65 Iterations aufgerufen.

Anzahl der Threads \ Test	1	2	3
1	52.028838	52.140695	52.938863
2	26.087554	26.821407	26.110185
3	17.549110	17.515558	17.551143
4	13.151488	13.143059	13.140707
5	10.686832	10.544150	10.509162
6	8.777090	8.838507	8.783395
7	7.667535	7.544237	7.536613
8	6.610692	9.539265	9.666401
9	5.894046	6.146108	6.024277
10	7.688474	7.727260	7.723680
11	7.008712	6.315243	7.054828
12	5.211963	6.311325	4.857754

### 3 Partdiff-posix im Koordinatensystem



### 4 Interpretation der Ergebnisse

Die benötigte Zeit sinkt offensichtlich quadratisch (3). Außerdem schwanken die Testwerte je mehr Threads zum Einsatz kommen (2) und (3), was ungewöhnlich ist, da unser Schedulingverfahren statisch ist. Man kann außerdem annehmen, dass man anhand der Partdiff-posix mit 12 Threads einen Speedup  $\geq 8$  erhält, da der kleinste Testwert von Partdiff-seq mit 51.954834 und der größte Testwert von Partdiff-seq mit 6.311325 ein Speedup von  $\approx 8.232001$  erreicht.