

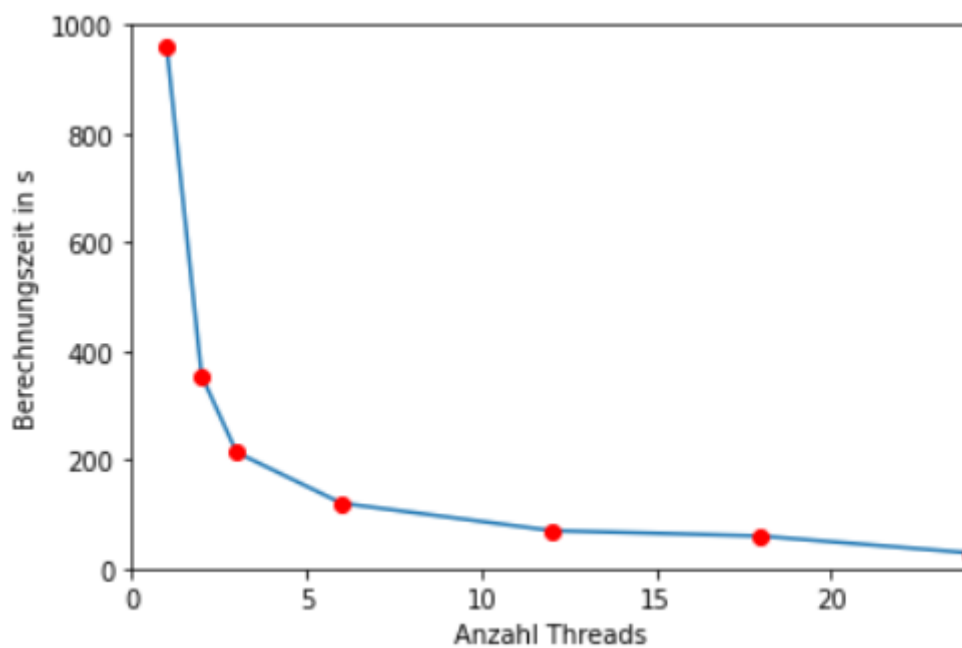
## Aufgabenblatt 6

---

### Benchmark:

. /partdiff t 2 4096 2 2 50

Threads	Berechnungszeit	+/-
1	961.578s	4.015s
2	354.675s	8.245s
3	213.700s	7.120s
6	120.154s	0.734s
12	69.820s	2.063s
18	59.604s	2.063s
24	28.595s	4.708s



### Hardware: ant13

```
NodeName=ant13 Arch=x86_64 CoresPerSocket=24
CPUAlloc=6 CPUTot=48 CPULoad=2.72
AvailableFeatures=(null)
ActiveFeatures=(null)
Gres=(null)
NodeAddr=ant13 NodeHostName=ant13 Version=20.11.8
OS=Linux 4.18.0-348.2.1.el8_5.x86_64 #1 SMP Tue Nov 16 14:42:35 UTC 2021
RealMemory=120536 AllocMem=12288 FreeMem=109075 Sockets=1 Boards=1
State=MIXED ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
Partitions=vl-parcio
BootTime=2021-11-22T12:56:23 SlurmdStartTime=2021-11-22T12:57:10
CfgTRES=cpu=48,mem=120536M,billing=48
```

```
AllocTRES=cpu=6,mem=12G
CapWatts=n/a
CurrentWatts=0 AveWatts=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Comment=(null)
```

### Interpretation:

Mit dem parallelen Programm und 24 Threads ergab sich gegenüber dem seriellen ein Speedup von 28,4. Allgemein lässt sich sagen, dass für dieses Intervall eine Erhöhung der Threadanzahl die Laufzeit verringert. Auffällig ist jedoch, dass die Unterschiede der Laufzeiten immer kleiner werden je mehr Threads genutzt werden. Zwischen 2 und 3 Threads ist der Unterschied 115 s, zwischen 10 und 11 nur noch 4,7 s und zwischen 23 und 24 Threads beträgt die Differenz der Laufzeiten gerade einmal 1,08 s. Erklären lässt sich dies durch die steigende Synchronisationsarbeit bei steigender Threadanzahl. Diese "dämpft" den Speedup, wodurch die Differenzen in der höheren Threadanzahl nicht mehr so groß, wie am Anfang sind. (→ Amdahls Law)