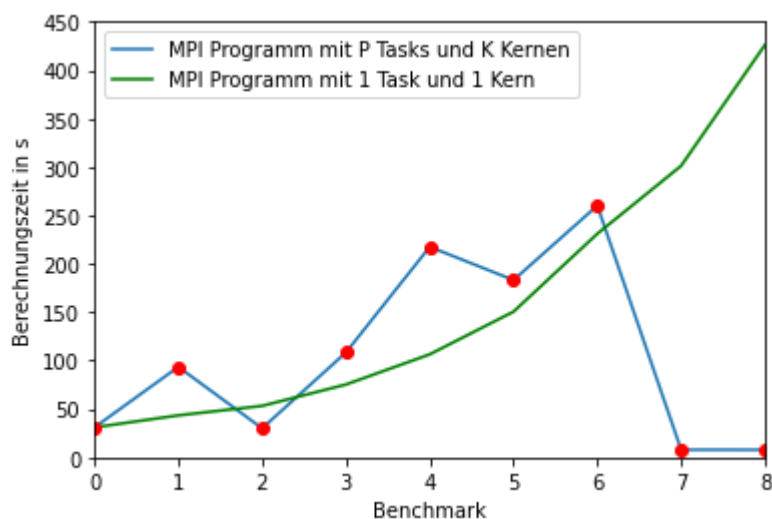


Aufgabenblatt 8

K	P	N	I	Berechnungszeit	+-	Berechnungszeit 1 Kern 1 Task
1	1	836	256	31.070s	0.027s	31.070s
1	2	1182	256	93.539s	0.248s	43.464s
1	3	1448	256	30.119s	0.060s	53.244s
1	6	2048	256	108.410s	0.434s	75.322s
1	12	2896	256	217.548s	1.159s	106.477s
1	24	4096	256	183.198s	1.615s	150.587s
2	48	5793	256	260.136s	2.880s	213.055s
4	96	8192	256	7.934s	0.132s	301.236s
8	192	11585	256	7.912s	0.133s	426.062s



Hardware: ant13 (hauptsächlich, außerdem ant14-20)

```

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

Die Werte lassen sich schlecht interpretieren, da sie keine wirklich einheitliche repräsentation bieten (die Zahl der Genauigkeit ändert sich bei jedem Benchmark). Würde man im Gegenzug alles nochmal für einen Task und einer Node laufen lassen (aber mit passender Genauigkeit) würde man besser die Parallelisierung messen können.

Das Diagramm zeigt, dass das Original-Programm (mit 1 Task auf einem Kern) eine Exponentiell steigende Berechnungszeit hat, während das MPI Programm unterschiedliche (größtenteils auch ineffizientere) Berechnungszeiten hat. Jedoch nur bis zu dem Punkt, wo man auf mehr als 2 Kernen zusätzlich rechnet.

MPI ist somit deutlich effizienter für größere Anwendungsbereiche, als für den kleinen Anwendungsbereich, wo dann pthreads oder MPI effizienter wären.