Csaw1585 ZanolinCamillo

Sheet 01

Exercise1:

- SLURM

Submit: sbatch <jobscript>
Alter: scontrol show/update <jobid>

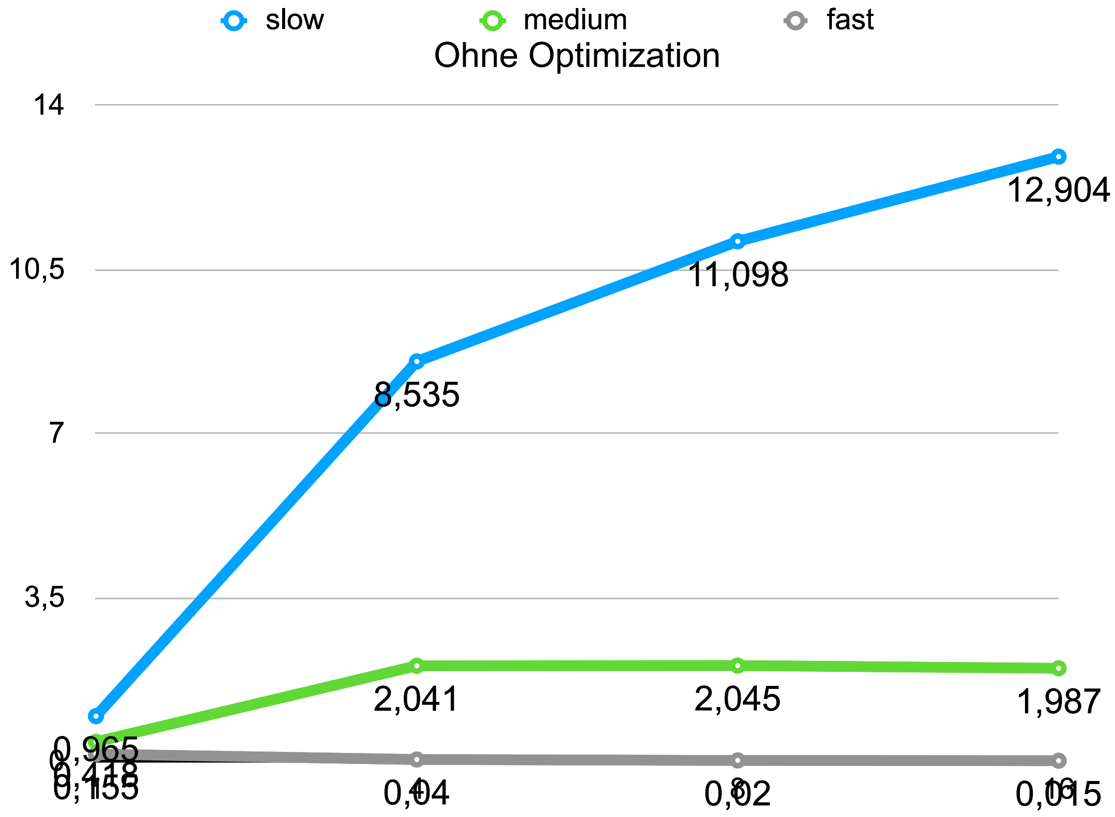
Check: squ for queued jobs in cluster
Cancel: scancel <jobid>

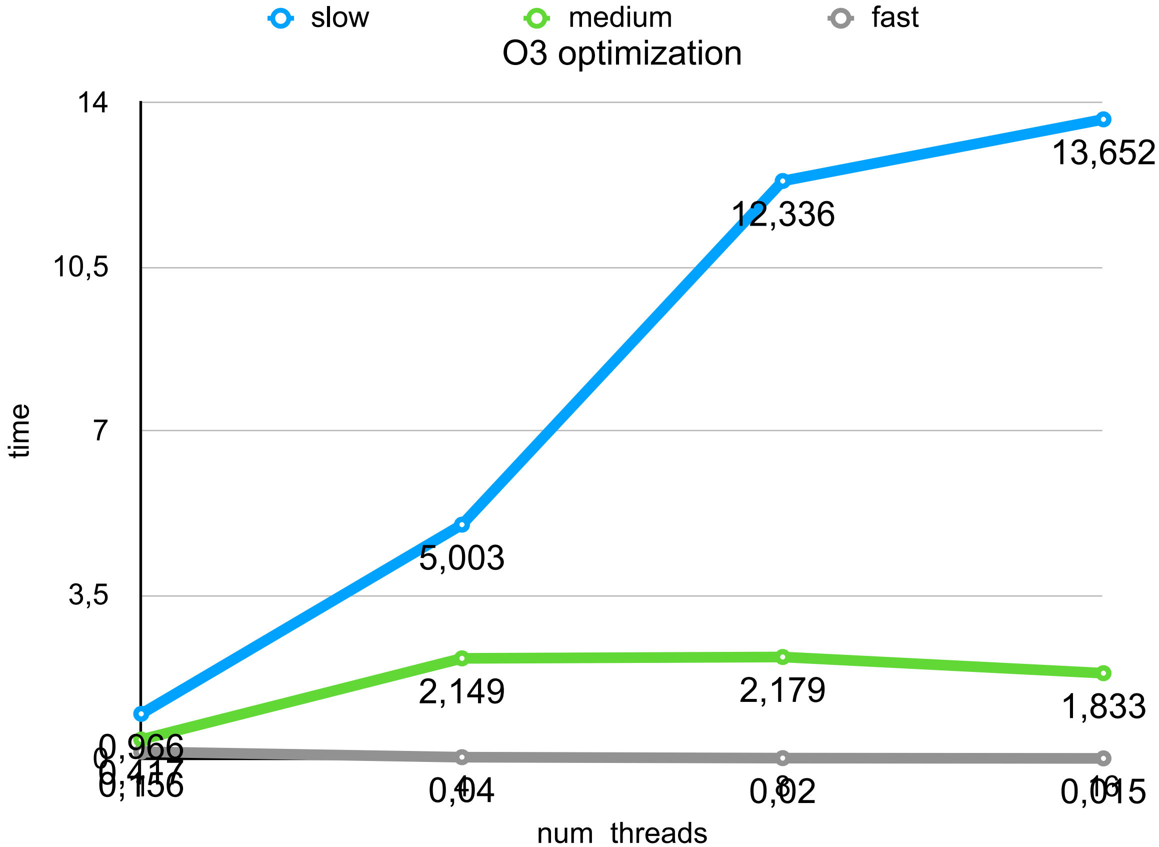
— job_name=name	To find and easily stop or quit a job. Find it in squeue output
—output= filename_pattern	To find the output easily so that bn
—time= time	To limit the time the job runs
—ntasks= ntasks	Requests the CPU resources to run ntasks Filling up on different nodes if not further specified
—mem=size (—mem-per-cpu)	To set the memory of the node! (in megabyte) Also possible to set per cpu (or gpu)
—cpus-per-task=ncpus	To tell slurm each task needs ncpus CPUs. Defaulting to one. To multithread more is needed.

- Jobs in parallel

You need MPI or openMP so that even a parallelization is possible. The compiler needs to support openmp directives (almost all do that). If thats the case the compiler flag -fopenmp needs to be set.

Exercise2:


slow
medium
fast
Ohne Optimization
14
10,5
7
3,5
0
12,904
11,098
8,535
2,041
2,045
1,987
0,965
0,418
0,155
0,04
0,02
0,015
1
4
8
16


slow
medium
fast
O3 optimization
14
10,5
7
3,5
0
time
13,652
12,336
5,003
2,149
2,179
1,833
0,966
0,417
0,156
0,04
0,02
0,015
1
4
8
16
num_threads

No good optimization can be reached with the flags. And it seems that the overhead of creating the threads is far greater than the benefit of using more threads. Except in the fast mode where it seems to be optimized for multiple openMP threads.

With multiple seems to be rather stable except in the fast one where one time an extraordinary time has been reached.

```
Executing fast with 1 threads
sum: 67108864, time: 0.0000 seconds

Executing fast with 4 threads
sum: 67108864, time: 0.0002 seconds

Executing fast with 8 threads
sum: 67108864, time: 0.0003 seconds

Executing fast with 16 threads
sum: 67108864, time: 0.0005 seconds
```