

I Parallelization of DFPT calculations

The `PHonon` package enables calculations of phonon eigenvectors and eigenmodes. This chapter examines the best ways to run `PHonon` calculations.

I.1 Optimal parallelization parameters for DFPT calculations

As discussed in sec. ??, the `PHonon` package offers the same three parallelization levels as the `PWscf` package, namely plane wave, k point and linear algebra parallelization. Furthermore parallelization on q points (so called image parallelization) can be used.

I.1.1 k point parallelization

In a first step, the same k point parallelization benchmark as in sec. ?? is run. This is depicted in fig. I.1.

Interestingly, the result from the `PWscf` calculation on silicon from sec. ?? is not reproduced here: the smallest pool size of 2 is not the one parallelizing best, but instead it is pool size 8. Furthermore, for more than 50 processors, even the biggest pool size 18 shows better scaling than the pool size 2. This is similar to the results in the `PWscf` benchmark with k point parallelization on the TaS_2 benchmarking system in sec. ??, as the separation between the different pool sizes isn't as clear as in the same benchmark on the silicon benchmarking system.

some more interpretation possible

The phonon benchmark has a similar runtime to the `PWscf` benchmark on TaS_2 shown in sec. ??, so comparison in wait time can reveal differences in the quality of parallelization between the two systems. Whereas the `PWscf` benchmark on TaS_2 had wait time not exceeding about 8 % of the wall time, the wait time shown in fig. I.2 between 10 % and 50 %. A possible explanation can be found how the time is actually spent during the calculation. In the case of the phonon calculation on silicon, the time of one iteration is on the scale of seconds, whereas one iteration for the `PWscf` calculation on TaS_2 is about 1 min.

some more interpretation

I.1.2 Linear algebra parallelization

Fig. I.3 shows that using linear algebra parallelization has so significant impact on the speedup, which is again in contrast to the results from sec. ??.

a bit more interpretation should be possible

I.1.3 Image parallelization

Better introduction

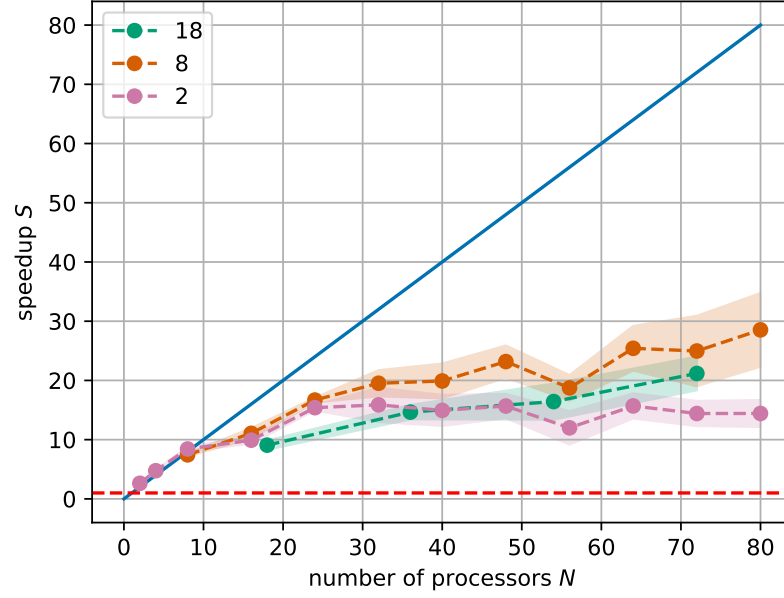


Figure I.1: Scalability utilizing k -point parallelization for the Si benchmarking system with three sizes of processor pools, QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, nd 1

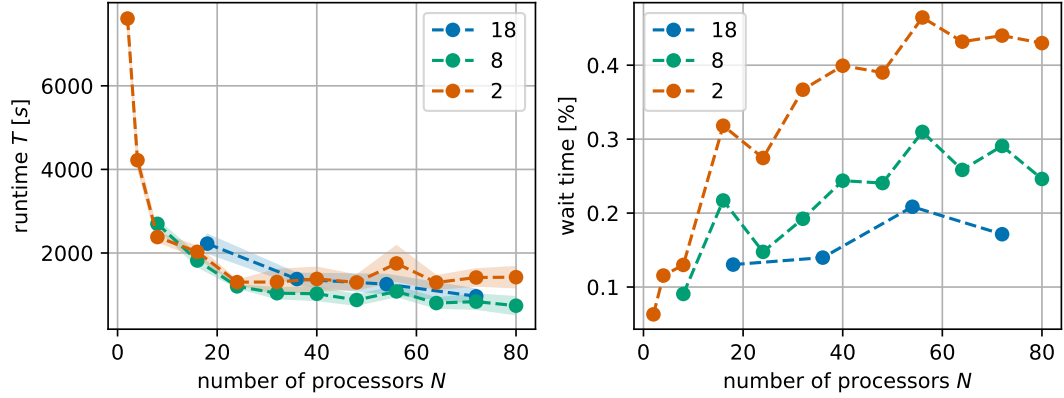


Figure I.2: Absolute runtime and wait time for the scalability test utilizing k -point parallelization for the Si benchmarking system with three sizes of processor pools, QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, nd 1

When using image parallelization, QUANTUM ESPRESSO outputs a separate time report for every image, so one step is added to the analysis: The total runtime of a calculation is determined by the longest running image, so speedup will be calculated using that value, but another important measure to evaluate is variation of times between images. This is depicted in fig. I.5.

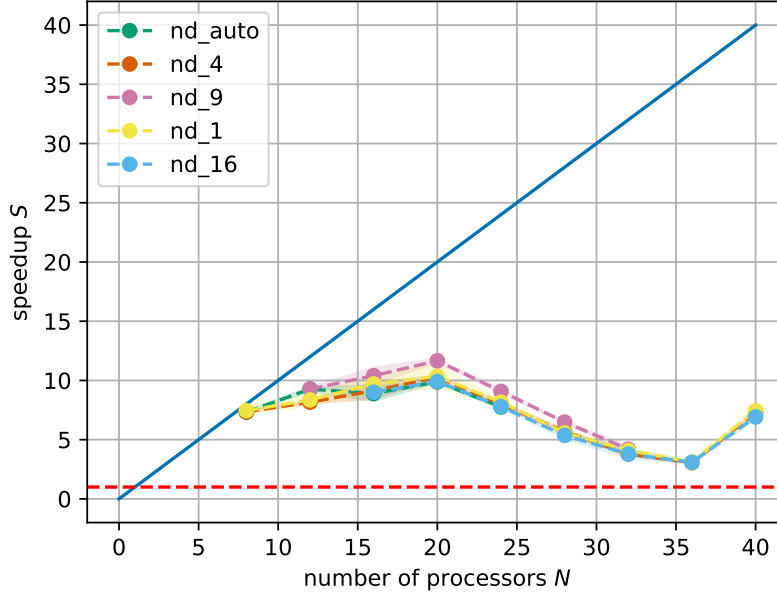


Figure I.3: Scalability utilizing linear algebra parallelization for the Si benchmarking system, QUANTUM ESPRESSO compiled with [Intel oneAPI 2021.4](#), `nk 1`

As the times between images don't vary much, good load balancing between images can be assumed for the silicon benchmarking system.

With the maximum time across images, speedup is then calculated, shown in fig. I.5.

I.2 Phonon calculations on TaS₂

The results from the last section can be used to estimate good parallelization parameters for a phonon calculation at the Γ point for TaS₂ in the charge density wave phase. The calculations were run on 180 processors, once with the previous established optimal pool size of 36 and once with a pool size of 18 for comparison. The relevant benchmark values for this calculation are listed in tab. I.1.

Table I.1: *CAPTION*

| | runtime | wait time |
|--------------|----------|-----------|
| pool size 18 | 3044 min | 0.16 |
| pool size 36 | 2020 min | 0.074 |

In this calculation the need for a good choice of parallelization parameters becomes especially clear: on the on the same number of processors, with the only difference in the choice of the parameter `nk`, the two calculations have a difference of 17 h.

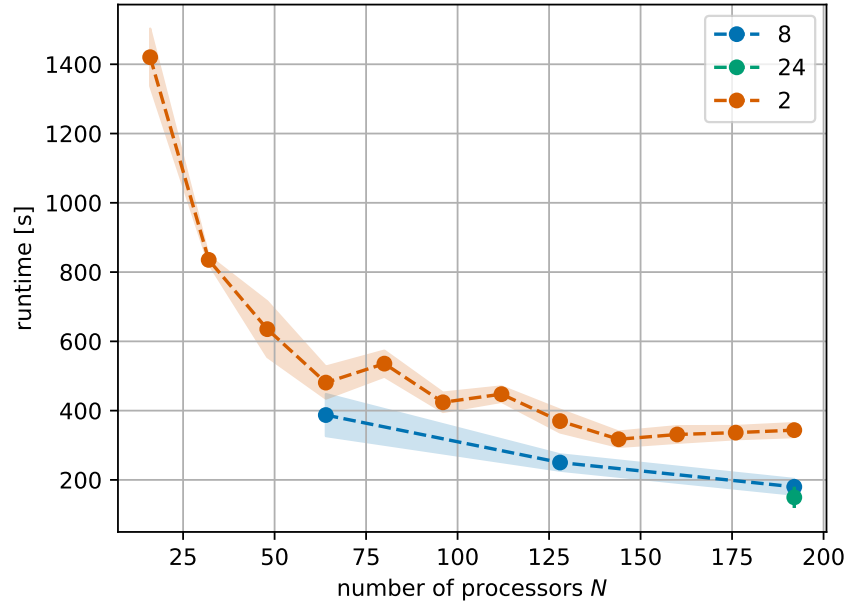


Figure I.4: Average runtime across images for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of n_i , QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, n_k , n_i chosen such that $\text{poolsize} = 8$, $n_d = 1$

I.3 Conclusion: Parameters for optimal scaling

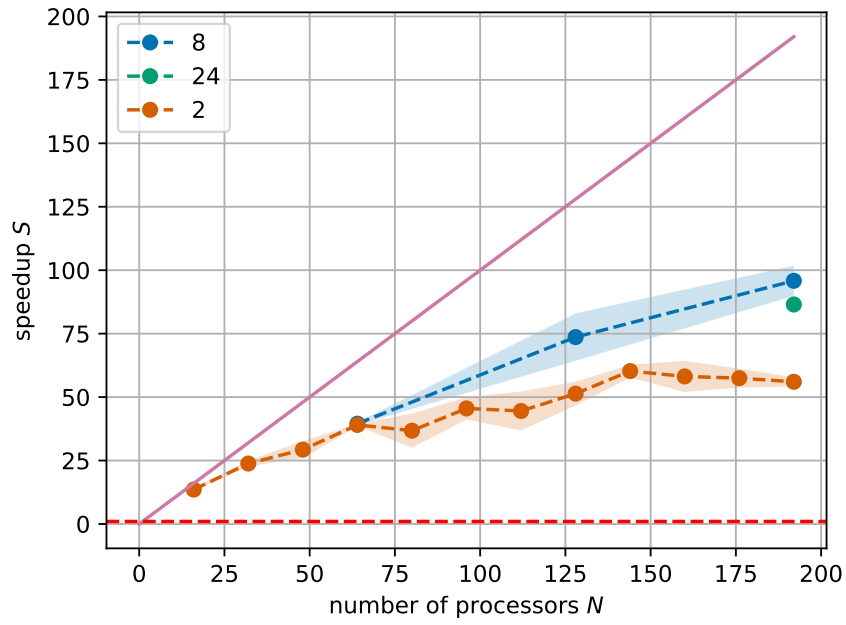


Figure I.5: Speedup calculated from the longest running image for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of n_i , QUANTUM ESPRESSO compiled with [Intel oneAPI 2021.4](#), n_k , n_i chosen such that poolsize = 8, n_d 1

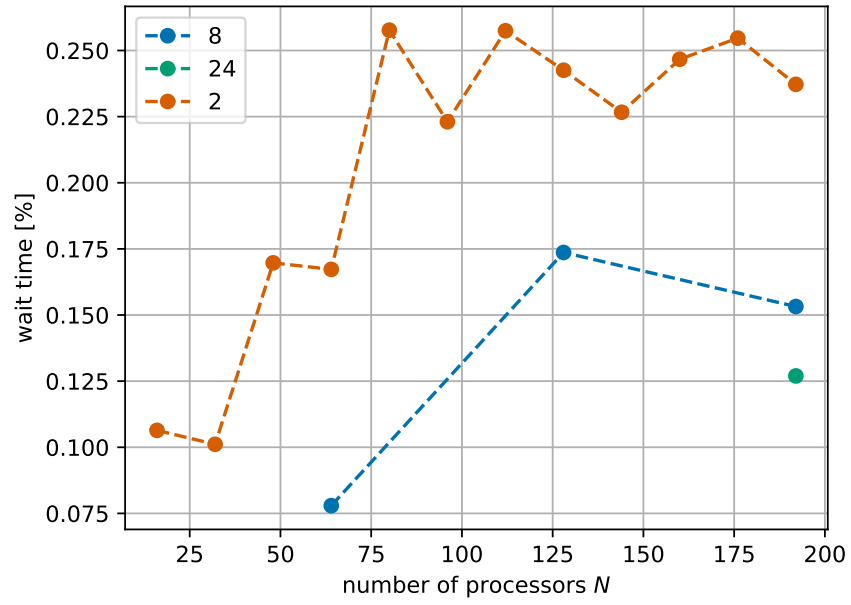


Figure I.6: Wait time calculated from the longest running image for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of ni , QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, nk , ni chosen such that poolsize = 8, nd 1