I Parallelization of phonon calculations

The `PHonon` package enables calculations of phonon frequencies and eigenmodes. This chapter examines the best ways to run `PHonon` calculations. The benchmarks are averaged over 10 runs.

I.1 Optimal parallelization parameters for phonon 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. The general picture is similar to `PWscf` benchmark with k point parallelization on the TaS_2 benchmarking system in sec. **??**, as there isn't an optimal pool size over the whole range of processors, instead some pool sizes seem to work best for some ranges.

One possible conclusion to draw from ch. **??** is that the longer runtime result in the calculation profiting more from parallelization and as such also from bigger pool sizes. The phonon benchmark has a similar runtime to the `PWscf` benchmark on TaS_2 shown in sec. **??**, so a similar scaling behavior should be expected. Comparison in wait time reveals the differences in the quality of parallelization between the two systems, which results in the observed different scaling. 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 for these differences between the two kinds of calculation can be found in how the time is actually spent during the calculation (which can be found in the QUANTUM ESPRESSO output files): 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. This means that the proportion of time spent on the distribution of data is bigger for the phonon calculation on silicon compared to the `PWscf` calculation on TaS_2, which introduces wait times.

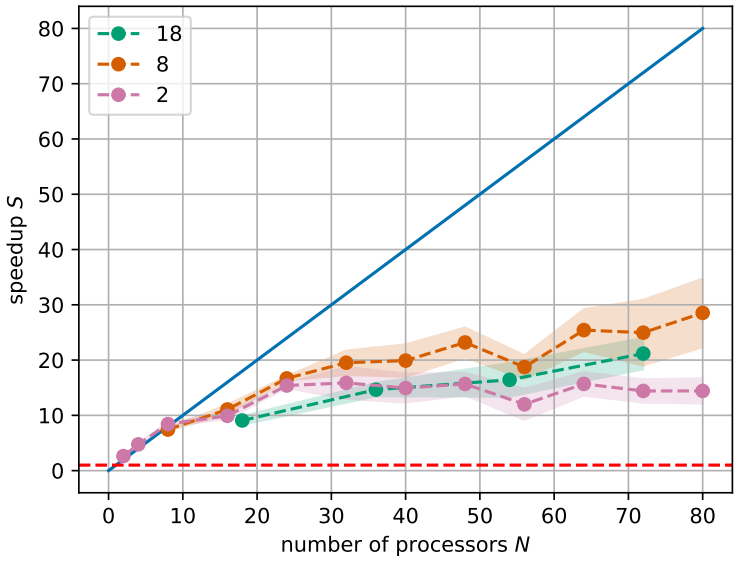

Figure I.1: *Scalability utilizing k-point parallelization for the Si benchmarking system with three sizes of processor pools,* QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nd 1`

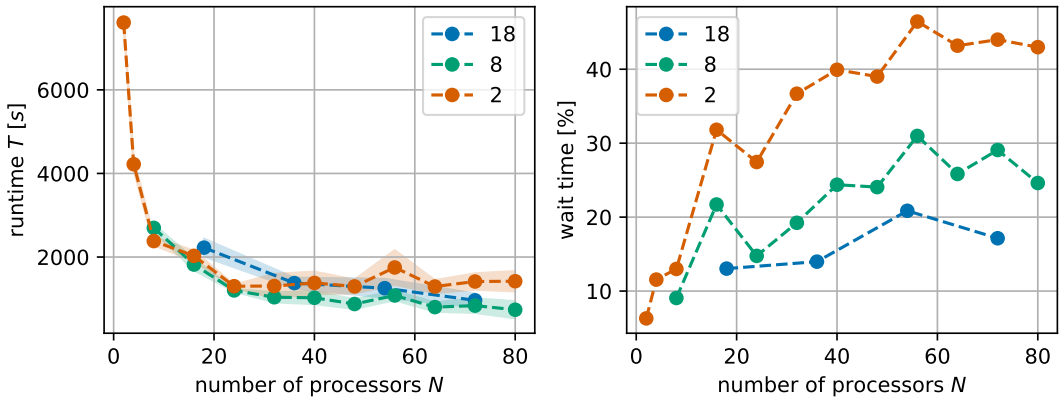

Figure I.2: *Absolute runtime and wait time for the scalability test utilizing k-point parallelization for the Si benchmarking system with three sizes of processor pools,* QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nd 1`

From the result of this benchmark, the parameters for the image parallelization in sec. I.1.3 can be set: the pool size will be fixed at 8.

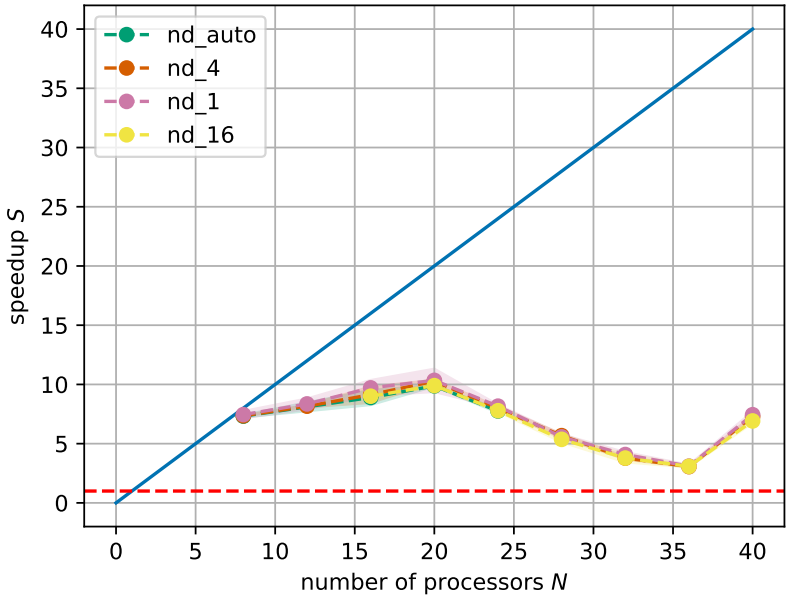

Figure I.3: *Scalability utilizing linear algebra parallelization for the Si benchmarking system,* Quantum ESPRESSO compiled with Intel oneAPI 2021.4, `nk 1`

I.1.2 Linear algebra parallelization

Fig. I.3 shows that using linear algebra parallelization has so significant impact on the speedup. This is again in contrast to the `PWscf` results from sec. **??**, where using linear algebra slowed down the calculation. This is again similar to the `PWscf` calculation on TaS_2.

As such, linear algebra parallelization will not be used in the benchmarks for image parallelization.

I.1.3 Image parallelization

When using image parallelization, Quantum ESPRESSO outputs a separate time report for every image, so one additional step is needed in the analysis: While the total runtime of a calculation is determined by the longest running image, the variation of times between images is important to judge how well the work between images is distributed. This is depicted in fig. I.5.

As the times between images are quite constant, 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.

The speedup shows that using image parallelization helps the phonon calculations scale over more processors than just using k point parallelization. Even just using 2 images almost

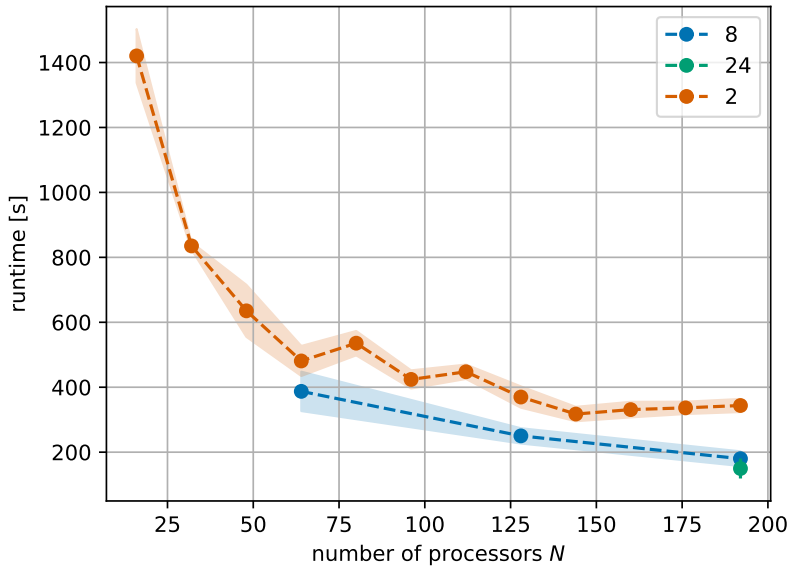

Figure I.4: *Average runtime across images for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of* `ni`, QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nk, ni` chosen such that poolsize = 8, `nd 1`

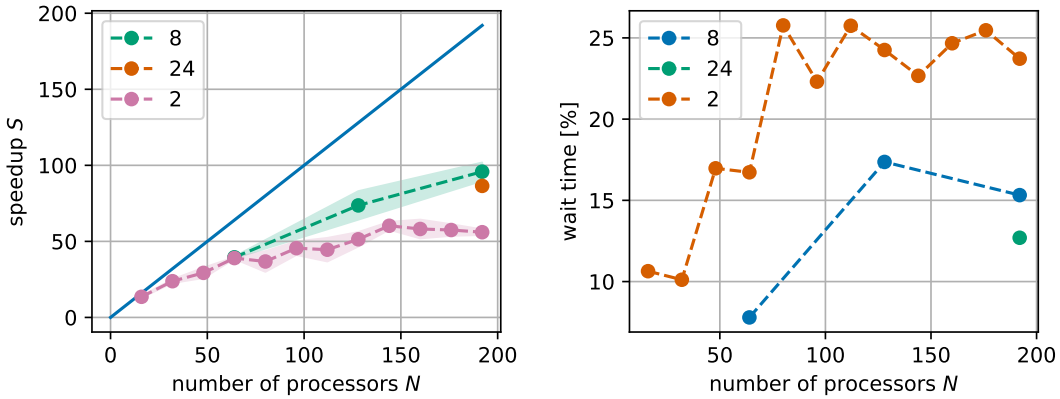

Figure I.5: *Speedup and wait time calculated from the longest running image for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of* `ni`, QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nk, ni` chosen such that poolsize = 8, `nd 1`

doubles the maximal achieved speedup in fig. I.1. This behavior doesn't extend across the whole range of processors, but using more images can partially elevate this problem.

The wait time (also calculated from the longest running image) shown in fig. I.2 is particularly good for every first data point, so in cases where only image parallelization without additional

k point parallelization is applied. As an example, for 64 processors, 8 images is ${}^{N_P}/_{N_i} = 8$, so the pool size is already 8 as required and the parameter `nk` is chosen to be 1. So at least for the silicon system it seems to be advisable to choose the parameter `ni` as large as possible, so that `nk` can be chosen as small as possible while keeping a pool size of 8.

I.2 Phonon calculations on TaS_2

The results from the last section can be used to estimate good parallelization parameters for a phonon calculation at the Γ point for TaS_2 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: *Benchmark values for phonon calculations of TaS_2 in the charge density wave phase, run on 180 processors*

	runtime	wait time
pool size 18	3044 min	16 %
pool size 36	2020 min	7.4 %

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.

I.3 Conclusion: Parameters for optimal scaling

The benchmarks on Quantum ESPRESSO's `PHonon` module showed that calculation on the silicon system profited more from bigger k point pools than `PWscf` calculations. Linear algebra parallelization was shown to have no impact on calculations.

Image parallelization in combination with k point parallelization lead to very good scaling across a wide range of processors. Best results were found when the processors were only split up by image parallelization.

Phonon calculations on TaS_2 in the charge density wave phase were then carried out, with the best time reached for a pool size of 36.