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.

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.

Generalizing from the benchmarks in ch. **??**, 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.

parameter choice and estimate for image parallelization

the pool size for testing linear algebra parallelization can be set to 8.

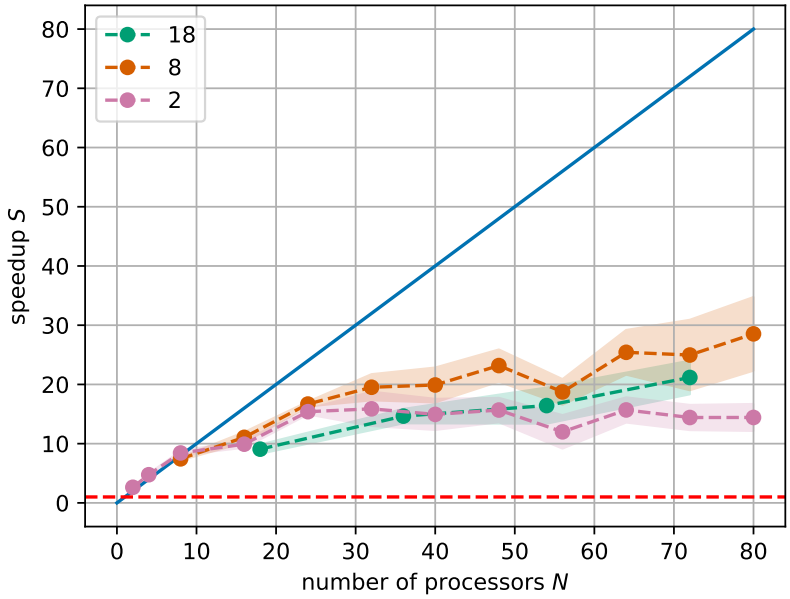

Figure I.1: *Scalability utilizing k-point parallelization for the Si benchmarking system with three sizes of processor pools,* QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nd 1`

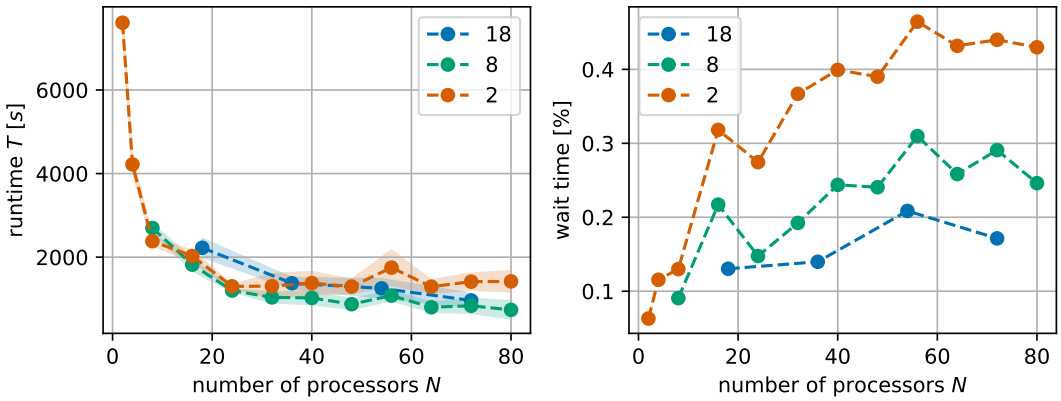

Figure I.2: *Absolute runtime and wait time for the scalability test utilizing k-point parallelization for the Si benchmarking system with three sizes of processor pools,* QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nd 1`

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

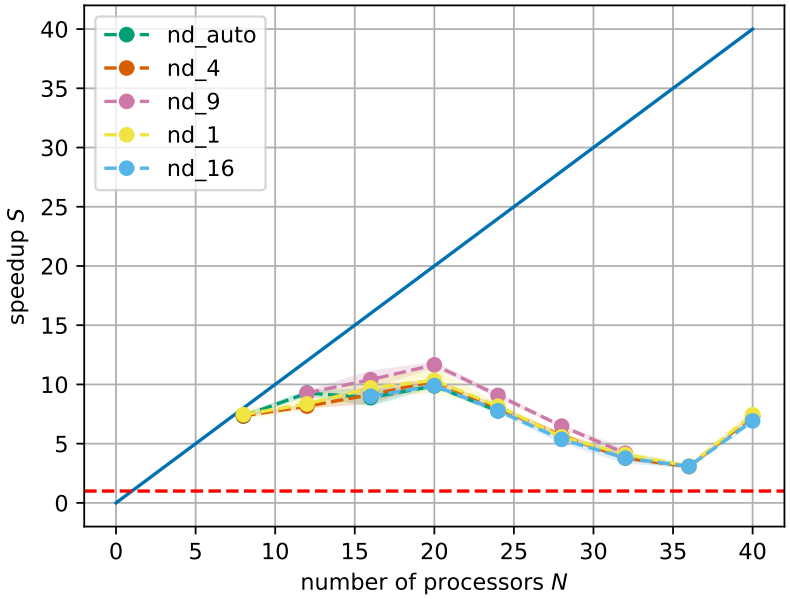

Figure I.3: *Scalability utilizing linear algebra parallelization for the Si benchmarking system,* Quantum ESPRESSO compiled with Intel oneAPI 2021.4, `nk 1`

down the calculation. Here, a linear algebra group size of 9 even speeds up the calculation a bit, while all other group sizes show the same scaling as linear algebra with `nd 1`.

As linear algebra parallelization is inconsequential to the form of the scaling, it 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 load balancing between images. This is depicted in fig. I.5.

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.

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

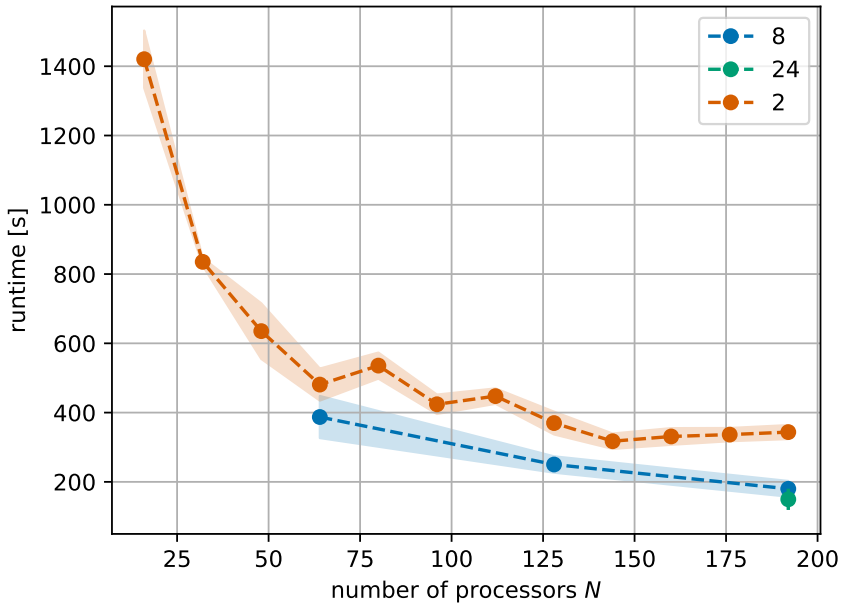

Figure I.4: *Average runtime across images for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of* `ni`, QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nk, ni` chosen such that poolsize = 8, `nd 1`

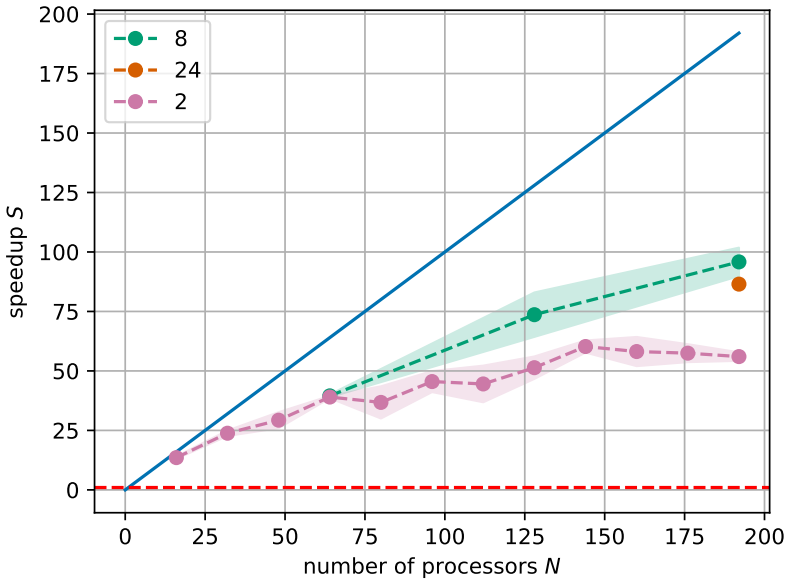

Figure I.5: *Speedup calculated from the longest running image for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of* `ni`, QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nk, ni` chosen such that poolsize = 8, `nd 1`

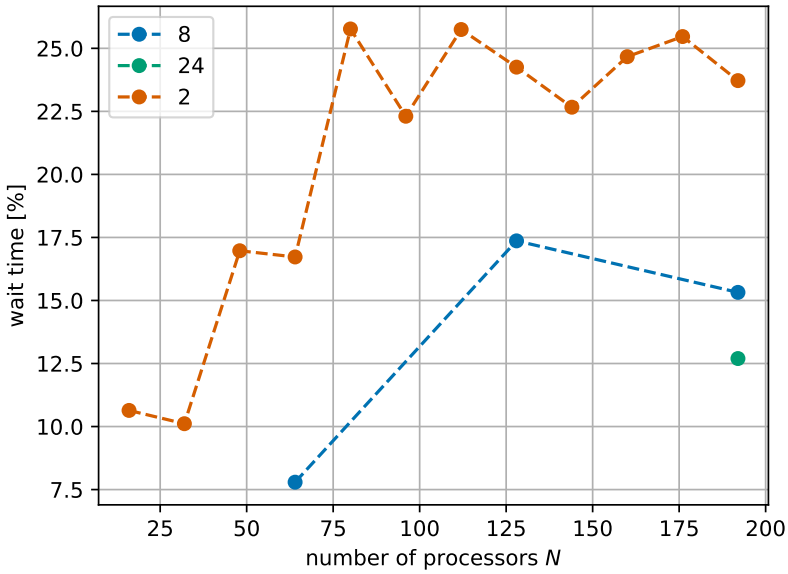

Figure I.6: *Wait time calculated from the longest running image for the scalability test utilizing image and k point parallelization on the Si benchmarking system with three values of* `ni`, QUANTUM ESPRESSO compiled with Intel oneAPI 2021.4, `nk, ni` chosen such that poolsize = 8, `nd 1`

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: *CAPTION*

	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