FAQ: Hybrid MPI and OpenMP/threaded programs

Performance of Hybrid Code

Before going into production run with a code which supports hybrid mode, either via OpenMP or via automatic parallelization, please check whether performance is not better running with one thread per MPI process. **Please note:** Altogether removing `-openmp` may improve performance of hybrid MPI+OpenMP codes (which then run as pure MPI codes): For these codes, if you are running with `OMP_NUM_THREADS` set to 1 because you want to run the "pure" MPI case, the performance of your code may be better if you compile/link your code without the `-openmp` flag. If you compile/link with the flag, the performance of your code may be penalized with the OpenMP overhead even though you don't want to use OpenMP since the compiler may produce less optimized code due to the OpenMP induced code transformations.

For codes that have explicit calls to OpenMP functions, either shield the calls with !$ directives, or compile them for the "pure" MPI case using the `-openmp_stubs` option instead of `-openmp`. A code compiled with `-openmp_stubs` will not work if `OMP_NUM_THREADS` is set to a value greater than 1.

Note that there may well be cases in which retaining hybrid functionality may give a performance advantage e.g. if your code becomes cache-bound and little shared-memory synchronization is required. But you need to check this, and optimize the number of threads used if you decide in favour of hybrid mode.