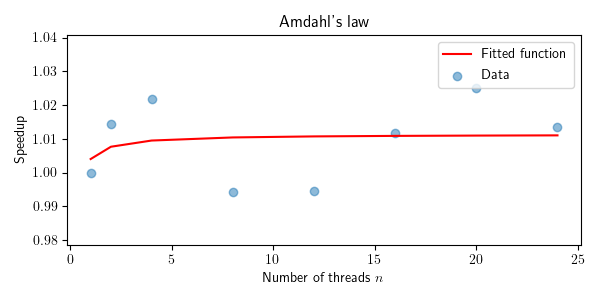
Speedup in HPC is defined as , where is the time taken by a task on core, and is the time taken on cores. Strong scaling is concerned with how the solution time varies with the number of processors for a fixed *total* problem size; weak scaling is concerned with how the solution time varies with the number of processors for a fixed problem size *per processor*.

Amdahl’s law governs strong scaling. It states that where is the proportion of the execution time spent on the serial part and is the proportion spent on the parallelised part. This means that, for a fixed problem, the upper limit of speedup is determined by the serial fraction of the code.

Hamilton uses nodes, each with CPUs. Each CPU has cores and threads. To ensure consistency, each test booked a whole node, but used only CPU. For both weak and strong scaling tests, and threads were used. For the strong tests, the same particle file was used; for the weak tests, the number of particles scaled with the number of threads.

All I/O operations were disabled.

Results are dispiriting, with , implying the code is *barely* parallelisable. This is likely a consequence of the test setup. Relevant code can be found in *parallel-report*. 6 SLURM scripts were generated using the *generate\_slurms.py* script. Each set a number of threads: 1, 2, 4, 8, 12, 16, 20, and 24. For each, a node was reserved, and an equal number of CPUs and threads were assigned to the task. All I/O output was disabled, except the writing of the duration to stdio. 100 particles were used for each, and a timestep of .



Weak scaling is concerned with how the solution time varies with the number of processors for a fixed problem size *per processor*. Weak scaling is governed by Gustafson’s law, which states: . It is based on two approximations, that the parallel part scales linearly with the amount of resources, and that the serial part does not increase with respect to the size of the problem.

**Testing**

16 different processes were run. 8 were strong, with only the number of cores varying – 1, 2, 4, 8, 12, 16, 20, and 24. The other 8 were weak, with the number of particles varying – 500, 600, 700, 800, 900, 1000, 1100, and 1200 and the number of cores

There is a severe overhead in simulations and noise on a system (don’t use small problem setups for measurements)

This may be a consequence of the small problem setup, as there are severe overhead in setting up a simulation.

I’ll try it with a fuck-off massive one shortly.

About 12 minutes for 1 THREAD and 100 particles.

Assuming it scales linearly… Fuck’s sake!