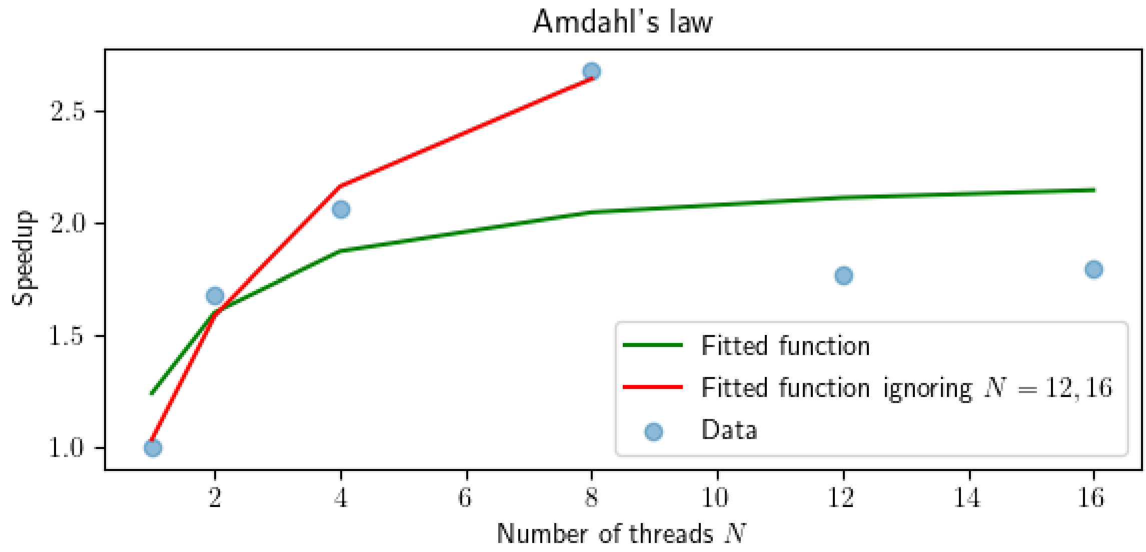
Speedup is defined as $\frac{t_1}{t_N}$, where t_1 is the time taken by a task on 1 core, and t_N is the time taken on N cores. Strong scaling is concerned with how the solution time varies with the number of processors for a fixed *total* problem size; weak scaling is concerned with how the solution time varies with the number of processors for a fixed problem size *per processor*. Amdahl's law governs strong scaling. It states that $speedup = \left(s + \frac{p}{N}\right)^{-1}$ where s is the proportion of the execution time spent on the serial part and p is the proportion spent on the parallelised part. Scalability was tested using Hamilton. Hamilton uses nodes, each with 24 CPUs. Each CPU has 8 cores and 16 threads. To ensure consistency, each test booked a whole node. For both weak and strong scaling tests, $1, 2, 4, 8, 12$, and 16 threads were used. For the strong tests, the same 100 particle file was used; for the weak tests, the number of particles scaled with the number of threads. All I/O output was disabled.

Results for strong scaling are dispiriting, with $s = 0.44$, implying the code is *barely* parallelisable. If the data points for $N = 12, 16$ are ignored, a better result of $s = 0.29$ is obtained. It is believed these anomalous results originate in a discrepancy between the hardware configuration of Hamilton and the SLURM scripts used.



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: $scaled\ speedup = s + p \times N$. It is based on two approximations, that the parallel part p scales linearly with the amount of resources, and that the serial part s does not increase with respect to the size of the problem. Weak scaling results are similarly disappointing, with $s = 1.06$, implying the code runs more slowly in parallel then serially. It seems, therefore, that a strong scaling model is more appropriate for this code.