Scaling Behavior for ROMS Code

October 6, 2020

To demonstrate our typical performance on SDSC's Comet computer, we report here two timing tests varying the number of compute nodes in the preceding figures. There are two plots for each model. The plot shows the reduction of wall clock time when increasing the number of CPUs (cores) in log-log axes. Ideally it should be a straight line with a slope of -1. The second plot shows computational speed per CPU as a function of the number of CPUs in log-linear axes. Ideal scaling should yield a constant; however, ROMS, as any practical code, exhibits first an increase of performance per node because of cache effects, followed by a degradation due to the increased relative cost of MPI communications and a decreased length of innermost loops (vector length) due to subdomains becoming smaller and smaller. In all the tests presented here we used a ROMS grid of $600 \times 500 \times 50$ points, OpenMPI IB version 1.8.4, Intel compiler v. 2016.1.210. The second plot provides a useful guideline about the "sweet spot" in terms of choosing the number of CPUs depending on the problem size for the optimal use of computing resources. On the other hand, concern for a prompt turnaround time may indicate using more cores at a slightly reduced efficiency.

As discussed in the Main Report, the WRF atmospheric model and the OASIS-MCT coupler are community codes with good scaling behavior on machines like Comet.

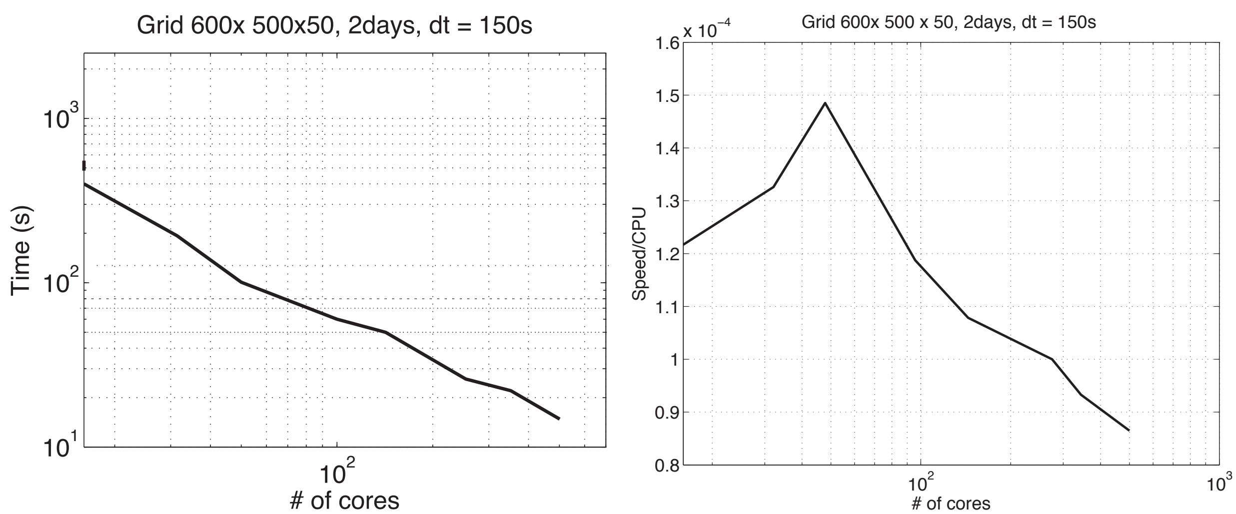

Figure 1: ROMS test