Pegasus2 computer system performance analysis based on Community Climate System Model (CCSM) simulation tests

Natalie Perlin, Joel Zysman, Benjamin Kirtman

Series of Community Climate Model System (CCSM) numerical simulations were conducted on Pegasus2 supercomputer in the University of Miami, to test performances of several MPI implementations, use of resources by the application, and its scalability. CCSM is a coupled general circulation global model interconnecting several model components, namely: atmospheric, ocean, land, and ice models. CCSM is used to simulate and predict the mean climate, as well as to estimate climate sensitivity and change due to various factors. CCSM is computationally demanding application targeting high-performance multiprocessing computations using Message Passing Interface (MPI).

Based on the CCSM model simulations timing results for 64 cpus, recent version (1.7.5) of open-source openmpi build with Intel compilers and Intel-build MPI implementation impi were found to yield about 30% increase in computation speed, as compared to the earlier versions of openmpi builds (Fig. 1). Further, Intel impi was found to be superior of all other MPI implementations for 256 and greater number of cpu-s. Allinea© PerformanceReport tool used for “bottleneck” analysis indicated notable increase in time spent for message-passing with the increased number of processors, leading to the loss in efficiency of overall compute time (Fig. 2). Scaling capacity of the given model application, being sub-linear due to progressive message-passing costs, was found to be systematically well approximated using the power fit (Fig.3). Upper limit of a number of cpu-s for a given application could then be rationalized. Application scalability depends both on system configuration and application configurations.

(10/2014)
Figure 1. CPU-time reported for the CCSM simulations, compiled with the Intel Fortran compiler (ifort) but employing different MPI implementations. All the tests were performed using 64 cores involving 4 nodes (16-core nodes), and series of 6 tests were conducted for the each MPI version. See text for more details on the CCSM run settings.

Figure 2. Breakdown of the total time of a simulation into the time spend for numerical calculations (CPU), message-passing (MPI), and input/output data (I/O), based on the Allinea Software © Performance Reports. Test were conducted for two MPI implementations, openmpi/1.7.5 and impi, and using up to 64 processors. Results reported are averages of the series of 6 tests per each combination.
Figure 3. Scaling capacity $C(N)$ from CCSM tests with two MPI implementations, estimated as normalized model throughput estimated as $C(N) = Y(N)/Y(1)$, where $N$ is the number of cpu-s, $Y(1)$ and $Y(N)$ are model throughput estimates from the simulations using 1 and $N$ processors, respectively. Note the logarithmic scale on both axes. Long-dash line marks linear scaling, and short-dash marks a power-fit line as indicated in the legend, as a function of number of processors. Power line fit coefficients (1.28 and 0.77) are determined empirically. In all the tests except for N=1024 cpu-s, throughput was averaged for 6 identically set simulations.