Accelerated Prediction of the Polar Ice and Global Ocean (APPIGO)

Eric Chassignet
Center for Ocean-Atmosphere Prediction Studies (COAPS)
Florida State University, PO Box 3062840
Tallahassee, FL 32306-2840
phone: (850) 645-7288 fax: (850) 644-4841 email: echassignet@fsu.edu

Award Number: N00014-13-1-0861
https://www.earthsystemcog.org/projects/espc-appigo/

Other investigators:

Phil Jones (co-PI)	Los Alamos National Laboratory (LANL)
Rob Aulwes	Los Alamos National Laboratory
Tim Campbell	Naval Research Lab, Stennis Space Center (NRL-SSC)
Mohamed Iskandarani	University of Miami
Elizabeth Hunke	Los Alamos National Laboratory
Ben Kirtman	University of Miami
Alan Wallcraft	Naval Research Lab, Stennis Space Center

LONG-TERM GOALS

Arctic change and reductions in sea ice are impacting Arctic communities and are leading to increased commercial activity in the Arctic. Improved forecasts will be needed at a variety of timescales to support Arctic operations and infrastructure decisions. Increased resolution and ensemble forecasts will require significant computational capability. At the same time, high performance computing architectures are changing in response to power and cooling limitations, adding more cores per chip and using Graphics Processing Units (GPUs) as computational accelerators. This project will improve Arctic forecast capability by modifying component models to better utilize new computational architectures. Specifically, we will focus on the Los Alamos Sea Ice Model (CICE), the HYbrid Coordinate Ocean Model (HYCOM) and the Wavewatch III models and optimize each model on both GPU-accelerated and MIC-based architectures. These codes form the ocean and sea ice components of the Navy's Arctic Cap Nowcast/Forecast System (ACNFS) and the Navy Global Ocean Forecasting System (GOFS), with the latter scheduled to include a coupled Wavewatch III by 2016. This work will contribute to improved Arctic forecasts and the Arctic ice prediction demonstration project for the Earth System Prediction Capability (ESPC).

OBJECTIVES

The objective of this effort is to create versions of the Los Alamos Sea Ice Model (CICE), the HYbrid Coordinate Ocean Model (HYCOM) and the Wavewatch III models that can perform optimally on both GPU-accelerated and MIC-based computer architectures. These codes form the ocean and sea ice components of the Navy's Arctic Cap Nowcast/Forecast System (ACNFS) and the Navy Global Ocean Forecasting System (GOFS), with the latter scheduled to include a coupled Wavewatch III by 2016.

This work will contribute to improved Arctic forecasts and the Arctic ice prediction demonstration project for the Earth System Prediction Capability (ESPC).

APPROACH

We will utilize an incremental acceleration approach to ensure we maintain code fidelity while improving performance. We will begin by improving the performance of selected sections of each code and expanding those regions until we have accelerated the three application codes. Acceleration may start with directive-based mechanisms like OpenACC and OpenMP, but may also include targeted kernels written in CUDA or other lower-level accelerator libraries. This approach provides early successes and opportunities to test the changes as they are made. A second approach will redesign code infrastructure to incorporate a multi-level parallelism by design. The modified codes will be validated both on a single component basis and within the forecast systems.

WORK COMPLETED

Over the past year, we made progress in a number of areas to improve performance.

APPIGO and OLCF Hackathon:
The year began with participation in an Oak Ridge-sponsored “hackathon” to try and enable GPU acceleration of HYCOM and CICE on the Titan machine in collaboration with vendor representatives and Oak Ridge Leadership Computing Facility (OLCF) staff. Nearly the entire APPIGO team participated in this event.

HYCOM work during the hackathon was performed by Mohamed Iskandarani, Louis Vernon, and Alan Wallcraft with mentors Eric Dolven (Cray) and Carl Ponder (NVIDIA). Eric and Carl and Louis worked on OpenACC in HYCOM before the hackathon, and we had 4.5 days to get HYCOM running with OpenACC on a Cray XK7. Four major subroutines were ported (momtum, cnuity, mxkprf, hybgen). The mxkprf and hybgen subroutines are "column physics" routines, and required getting the compilers to in-line nested subroutines. Memory transfers between host and device were not optimized, and so HYCOM was 10x slower on the K20X Keplers than on the 16-Core AMDs alone. The initial lack of performance was not a surprise. Our goal was to identify any generic issues that needed work, and that was accomplished. The way forward is clear: explicitly leave all arrays in device memory (this is under development) and then minimize data transfers for (on host) halo exchanges. One major issue that was uncovered is that HYCOM's current "on error" behavior is to write an error message to the model log (stdout) and call MPI_Abort. This is entirely local to a single MPI task, but attached processors have no I/O capability and so can't generate error messages locally and can't issue an MPI_Abort. We will need to update HYCOM's error handler to allow for this.

At the hackathon Mohamed Iskandarani refactored subroutine barotp, and in particular looked at replacing the standard land/sea mask with not skipping land at all but zapping land, as needed, using a REAL array as a multiplicative mask with 1.0 for sea and 0.0 for land. He found that the zap approach was 10% faster, presumably because removing the mask gave better code optimization. However, this improvement was not maintained when running on the Intel Xeon processors of a Cray XC30 using the Intel compiler. We will continue to explore this optimization, but it is not currently a clear win and it introduces some additional complexity to the code.

Work on the CICE model during the hackathon was focused on a new refactored column physics package so that the improvements obtained could be applied to both the existing CICE model and the next-generation MPAS-CICE model. Despite efforts by Rob Aulwes and Doug Jacobsen (LANL) and the vendor and OLCF representatives, it was not possible to create an OpenACC version of the CICE column physics for acceleration. Current compiler technology could not apply OpenACC to the deep call tree inherent in the column physics. Attention has now returned to accelerating the dynamics and transport. An important technique for this portion of CICE will be handling the necessary boundary/halo updates. Rob has created a small test code to test GPUDirect on Titan that would enable the accelerator devices to access the network directly without moving data through the host.

HYCOM performance improvements

At NRL-SSC, Alan Wallcraft updated the standard 0.04 degree global HYCOM benchmark, used in the HPCMP "TI" benchmark suite, to HYCOM 2.2.98 which includes land masks in place of do-loop land avoidance. Results are shown in the figures below. Figure 1 reports total core hours on the y-axis, so a horizontal line would be perfect scaling. HYCOM is actually super-scalar out to 4000 cores and takes the same number of core hours on 1000 and 16000 XC40 cores. The Cray XC40, at ARL DSRC, has the latest generation of Intel Xeon processors, with AVX2 operations, but has a lower clock speed than the Cray XC30 at Navy DSRC. This is the 3rd generation in a row where HYCOM per core performance has not significantly changed, but the number of cores per node has increased (and cost per core decreased).

Figure 2 compares HYCOM 2.2.98 to the old 2.2.27 version, with static (common block) memory allocation and do-loop land avoidance. The blue curve is the same result as in the 1st figure, with 2.2.98 super-scalar from 1000 to 4000 cores. The red curve is 2.2.27 which performs about the same as 2.2.98 on 1000 cores but shows a gradual loss of scaling on more cores. The 2.2.27 result is not bad, but 2.2.98 is much better and can achieve an overall cost reduction of ~20% on larger core counts.

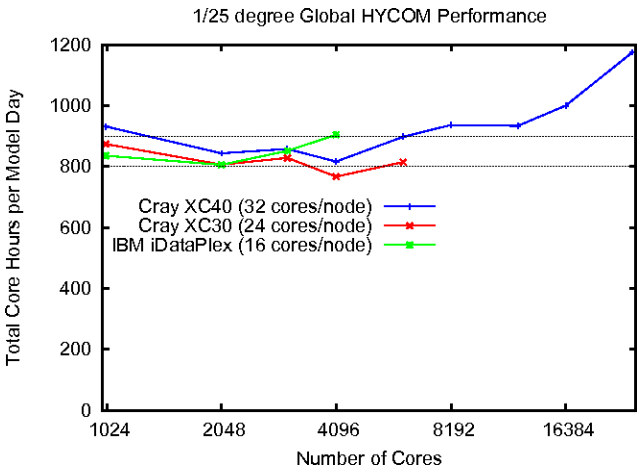

Figure 1. Core-hours per simulated model day as a function of core count for HYCOM on three machines.

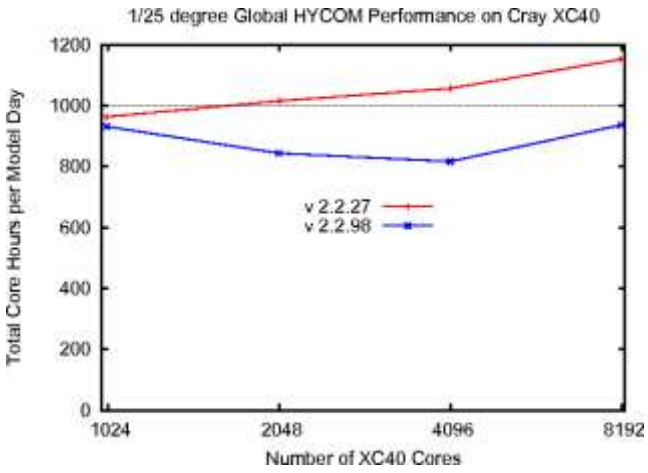

Figure 2. Core hours as a function of core count for two different HYCOM versions on a Cray XC40.

Additional performance improvements focused on vectorization and memory alignment important for both the Intel Phi architecture and GPU accelerators. The standard gx1v6 HYCOM benchmark has been run in native mode on 48 cores of a single 5120D Phi attached to the Navy DSRC's Cray XC30. Without optimization, individual subroutines run between 6 and 13 times slower than on 24 cores of a single Xeon E5-2697v2 node, and is overall 10 times slower. This is an indication that the code is not vectorizing well, but it isn't clear if this deficiency will carry over to the next generation Phi (with a much different core). I/O is also very slow.

During the AOLI program meeting in November, one approach suggested by Tom Henderson (NOAA) and tested in WRF is to use a fixed chuncksize of 4 or 8 as the first array dimension for column physics routines. Here 4 or 8 is the vector length for 64-bit REAL's under Intel AVX variants. Similarly, Alan Wallcraft attended a 2-day class at Stennis on the Intel Phi where the need for aligned arrays was stressed. This may mean that padding HYCOM's halo size from 6 to 8 elements may help, providing we also set the tile width (in memory) to a multiple of 8. Note that the array padding only changes how the arrays are laid out in memory, not the operation count. However, an 8-wide halo would reduce the number of MPI operations in the 2-D barotropic equation step.

We used the information above, together with other memory optimizations, to tune the standard 0.04 degree global HYCOM benchmark case described above for the Cray XC40 and the Intel fortran compiler. At Navy DSRC, the Cray XC40 has the latest generation of Intel Xeon processors, with advanced vector (AVX2) operations, but has a lower clock speed than the Cray XC30. For HYCOM, we setup the Intel fortran compiler for bit for bit reproducible results across any number of cores (ifort -fp-model precise -no-fma). We use fp-model precise because vector and scalar operations have different rounding, so the start and end of loop extents can't be scalar if the middle is vector. The no-fma option is only active with AVX2 instructions, which include fused multiply-add which has with different rounding than separate multiply add operations. On the Cray XC40, using huge pages improves performance by about 3%. In addition, making the first dimension of all arrays a multiple of 8 saved 3-6%. This requires changing a single number in the run-time patch.input file, and we run with "ifort -align array64byte" to align arrays on 64-byte boundaries. For example, we get a 10% improvement on the Cray XC40 for 8160 cores:

Standard: 6.1 minutes per model day
Multiple of 8: 5.7 minutes per model day
Huge pages: 5.5 minutes per model day

CICE Improvements

As described above, early attempts at GPU-acceleration of the CICE physics package were hampered by the evolving state of compiler technology and was unable to accelerate codes with deep call trees. While waiting for this software to improve, we (Rob Aulwes) moved toward accelerating the CICE dynamics package on the GPU using OpenACC. The dynamics package makes numerous message-passing (MPI) calls to perform halo updates and communicate neighbor information between nodes. This poses a significant obstacle to improving performance with GPU, since data would have to be copied from the GPU device to host memory, then transferred with MPI, and copied back to the device memory on the recipient host. A new approach called GPUDirect now provides a mechanism that works directly with the network Infiniband interconnect to bypass host memory and uses RDMA to directly copy data from the GPU device. Nividia has given Rob access to one of their clusters for him to test the GPUDirect code. Rob presented his work at Nvidia's GPU Technology Conference on March 19, 2015. Our initial implementation did not show improvement. This is primarily due to the current design of the existing MPI halo updates. The current locations of the updates do not provide

much opportunity for overlapping the communication with computation, effectively rendering the updates as bulk synchronous. We have identified alternate schemes to improve MPI with GPUDirect. Our next step will be to aggregate the messages. Even with normal use of MPI, it is usually better performance to send a smaller number of large messages than sending many small messages.

Another problem encountered during CICE optimization was the overhead associated with kernel startup on GPU devices. Using the .4 degrees CICE test problem that runs with 60 MPI ranks, we have been using the Nvidia profiling tools to analyze performance. Initial code structures in CICE look similar to the following pseudo-code

```
do iblk = 1, nblocks
  call construct_fields(iblk, nx_block,ny_block, …)
  do n = 1, ncat
    call construct_fields(iblk, nx_block,  ny_block, &
                  tm(:,:,:,n,iblk) …)
  enddo
enddo
```

Within each call to a subroutine, a kernel or set of kernels are launched asynchronously on the GPU. In theory, the kernels should be able to execute concurrently as each iteration is independent. However, profiling showed a significant overhead cost with launching the kernels:

```
======== API calls:
Time(%)     Time     Calls       Avg       Min       Max  Name
 7.21%  432.23ms     45504  9.4980us  8.1670us  317.01us  cuLaunchKernel
```

The profiler showed that cuLaunchKernel was being called over 45K times. Consequently, we explored schemes to reduce this count by fusing kernels. Our strategy was to refactor the subroutines by pushing the outer loops into the routines themselves. With these changes, we were able to reduce the number of calls to around 3K and also reduced the time to 38ms from 432ms (7.21% down to 0.67% of API call time). Prior to these changes, the runtime of the p4 problem was about 13% slower than the baseline. After these changes, we are now within 3% of the baseline.

Similar to GPUDirect above, we have been receiving assistance from Nvidia engineers to improve sharing of GPU devices by more than one MPI rank on a node, as will be the case on next generation architectures soon to be installed. CUDA provides a service called Multi-Process Service (MPS). This service allows multiple MPI ranks to take advantage of the Hyper-Q capabilities on newer Kepler architectures. Hyper-Q allows multiple GPU kernels originating from different MPI processes to be processed concurrently. Previously, if multiple processes attempted to use a common GPU device, the GPU had to perform context switches between kernel execution. This work is being done on an Nividia cluster that provides nodes with multiple GPU devices per node. The next step is to make CICE aware of device location. On nodes with multiple GPU devices, it is important to bind an MPI process to a local GPU device to avoid added latencies if the process uses a device on a different socket.

As part of continuing CICE model development at LANL, Elizabeth Hunke had created a new single-column version of CICE physics to prepare for a transition to the new MPAS-based sea-ice model and

decouple the column physics from the underlying mesh used for dynamics and transport. We spent some time during the year upgrading CICE to this version for further APPIGO development.

Model validation

While the performance changes are proceeding, we (Chassignet, Bozec) are configuring a version of HYCOM and CICE within the Community Earth System Model (CESM) as a test case for validating the new optimized versions. Starting with version 2.2.35 of HYCOM, we first focused on the proper implementation of the routine responsible for the exchange of fluxes between HYCOM and the ice and atmospheric components. We then updated HYCOM to the latest version available (2.2.98) and connected the river transport component. Also, we implemented the latest HYCOM-CESM together with the latest version of ESMF (7.0.0beta) on the NAVY DSRC machines, Kilrain (IBM iDataPlex) and Shepard (Cray XC30). A series of 20-year experiments were performed to ensure that HYCOM-CESM performed as expected when coupled with CICE only (G compset). When comparing HYCOM-CESM to identical stand-alone HYCOM-CICE and POP-CESM simulations, we find that differences arise over the polar regions.

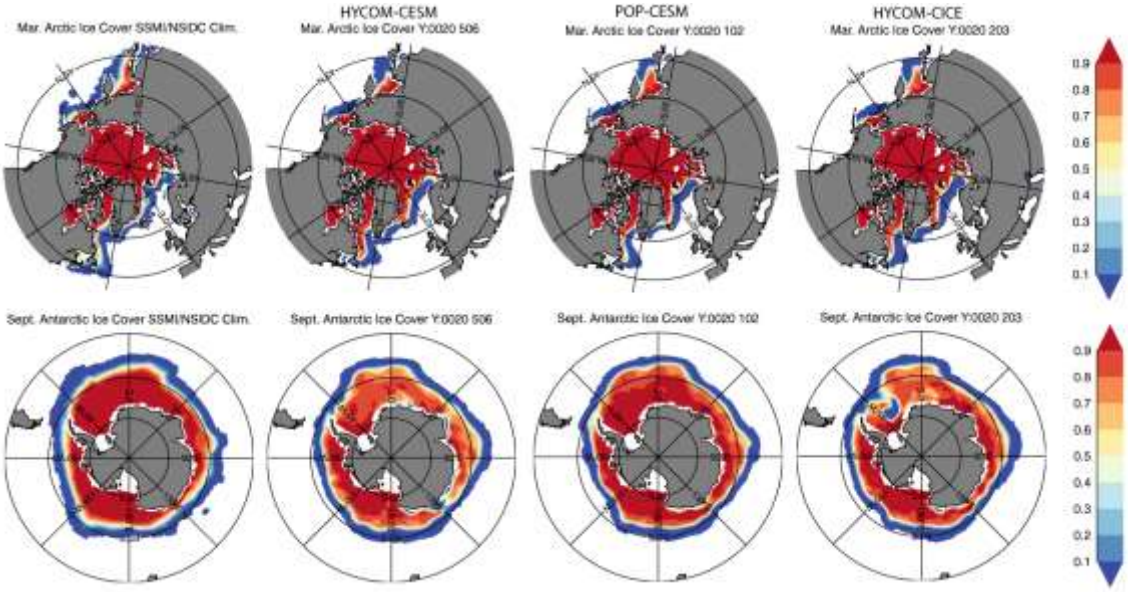

Figure 3: (Top) Ice cover in March (left to right) from NSIDC/SSMI climatology, HYCOM-CESM, POP-CESM and HYCOM-CICE at year 20 of the simulation. (Bottom) Same as above for September.

All three experiments show a reasonable ice extent compared with the climatological SSMI ice extent produced by the NSIDC in the Arctic region (Fig. 3, top). In the southern hemisphere, HYCOM-CESM exhibits a fully ice-covered Weddell Sea in September wersus a partial coverage in HYCOM-CICE (Fig. 3, bottom). Despite a better coverage, the ice cover extent of HYCOM-CESM is less than of POP-CESM or the SSMI climatology.

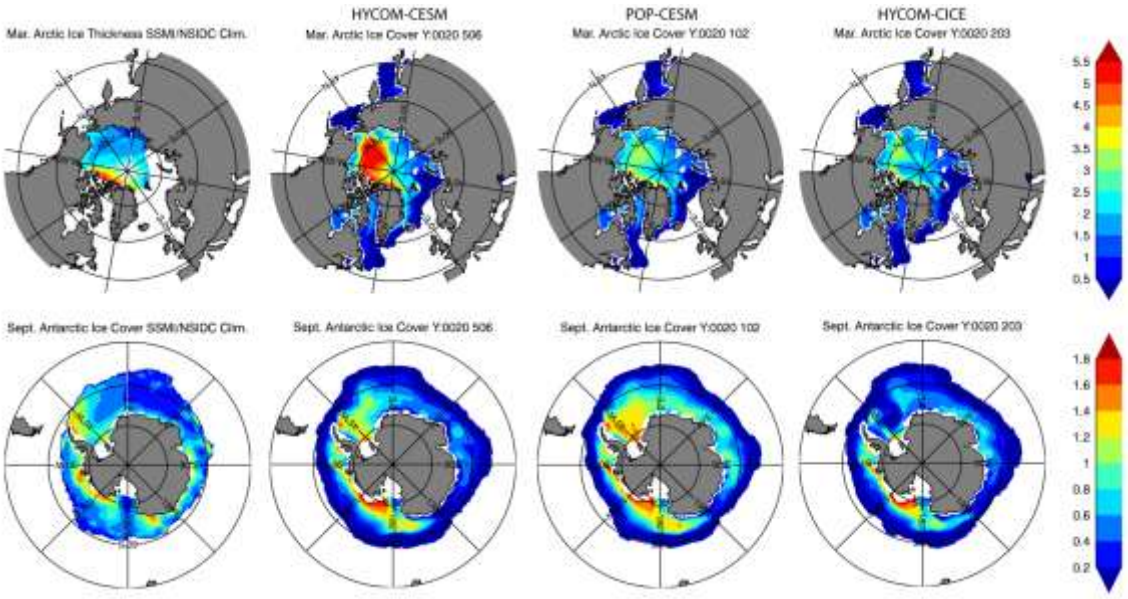

Figure 4: (Top) Ice Thickness in m in March (left to right) from NSIDC/IceSat climatology, HYCOM-CESM, POP-CESM and HYCOM-CICE at year 20 of the simulation. (Bottom) Same as above for September.

The main difference between the experiments is in the ice thickness. While the region of maximum ice thickness is limited to the northern coasts of the Canadian Archipelago in the NSIDC/IceSat climatology, the maximum ice thickness over the Beaufort gyre region is more pronounced in HYCOM-CESM (Fig. 4).

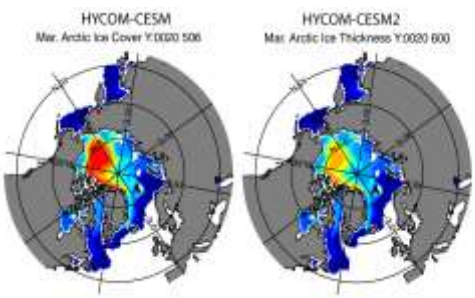

Figure 5: Ice Thickness in m in March (left to right) from HYCOM-CESM and HYCOM-CESM2 at year 20 of the simulation.

The time evolution of the total volume of ice over the Arctic region increases slightly in HYCOM-CESM over the 20 years with a seasonal cycle between 1.9×10^{13} m^3 and 3.5×10^{13} while it is stable in POP-CESM and HYCOM-CICE with a range of 1.0×10^{13} m^3 to 2.7×10^{13} m^3 and of 0.7×10^{13} m^3 to 2.6×10^{13} m^3, respectively (Fig. 6). To test the sensitivity of the ice to the ocean diffusion and viscosity parameters, a fourth experiment (HYCOM-CESM2) was performed with stronger coefficients as used for the CORE-II project. The maximum ice thickness is reduced to 4 m from 5 m in HYCOM-CESM (Fig.5). The time evolution of the total ice volume in HYCOM-CESM2 also stabilizes after 20 years with a weaker range than HYCOM-CESM. This result indicates a strong sensitivity of the ice thickness to the ocean circulation. In the southern ocean, POP-CESM has an overall higher ice thickness than in the HYCOM simulations, with maximum in the Weddell and Ross Seas (Fig. 4). We also notice a slight increase of the ice thickness in the Wedell Sea in HYCOM-CESM when compared to HYCOM-CICE (Fig. 4). We are currently investigating if these differences can be attributed to the differences in how fluxes are implemented over the ice in HYCOM-CESM and HYCOM-CICE.

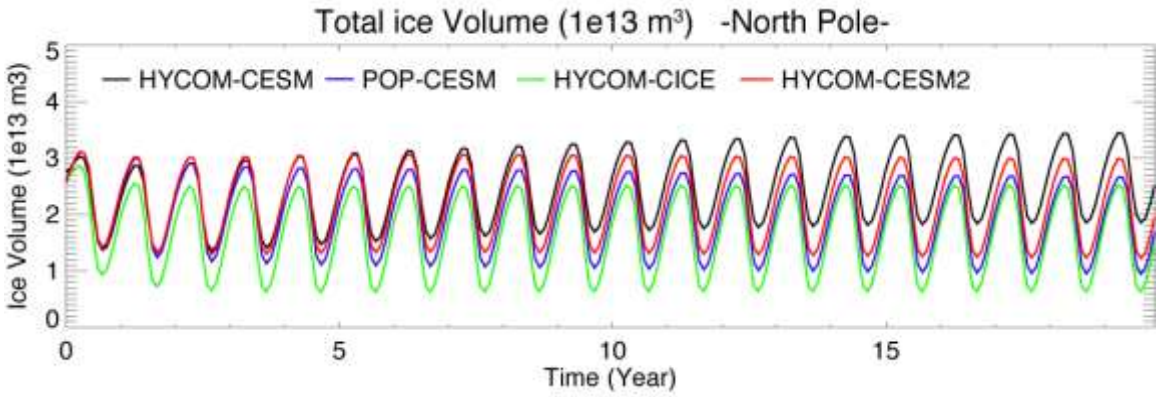

Figure 6: Evolution of the total ice volume in the Arctic Region for HYCOM-CESM (black), POP-CESM (blue) and HYCOM-CICE (green).

RESULTS

We continue to make some incremental improvements to HYCOM and CICE with vectorization, memory placement and message-passing optimizations. Larger improvements that fully exploit GPU accelerators or multi-core architectures like the Intel Phi remain elusive, in part due to immaturity of compilers and programming models and in part due to the current design of the two codes. However, we are learning a great deal about using the new systems and remain hopeful that significant improvements will be achievable.

Results have been presented at the annual program meetings and at Nvidia's GPU Technology Conference.

IMPACT/APPLICATIONS

Model performance improvements under this project will result in high-performance codes to enable improved future Arctic prediction, through improved resolution, increased realism or an ability to run ensembles.

RELATED PROJECTS

This project builds on the core model development activities taking place at the partner sites, including:

The Climate, Ocean and Sea Ice Modeling (COSIM) project that includes the primary development of the Los Alamos Sea Ice Model (CICE), funded by the US Department of Energy's Office of Science.

The ongoing development of the Arctic Cap Nowcast-Forecast System (ACNFS) and Global Ocean Forecast System (GOFS) at the Naval Research Lab – Stennis, funded by the US Navy.

Continued development of the Hybrid Coordinate Ocean Model (HYCOM) at Florida State University, funded by the National Science Foundation, Department of Energy and US Navy.