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

Eric P. 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: N00014-17-1-2159
https://www.earthsystemcog.org/projects/espc-appigo/

Other investigators:

Phil Jones (co-PI)	Los Alamos National Laboratory (LANL)
Rob Aulwes	Los Alamos National Laboratory
Alexandra Bozec	Center for Ocean-Atmosphere Prediction Studies
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. The APPIGO project is working to improve Arctic forecast capability by modifying component models to better utilize new computational architectures. We have focused on the Los Alamos Sea Ice Model (CICE) and the HYbrid Coordinate Ocean Model (HYCOM) and have been exploring and evaluating optimizations for each model on GPU-accelerated and many-core 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). We continue to make incremental improvements to these models and are focusing our effort on specific performance needs for the architectures planned for near-term deployment. We will implement new threading and task parallel approaches as well as vectorization improvements that can be completed and tested in the time frame of this project. These improvements are likely to be broadly applicable to a range of future architectures. The modified codes will be validated using a science-focused experiment in collaboration with related projects. This results from this project 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) and the HYbrid Coordinate Ocean Model (HYCOM) that can perform optimally on Intel many-core-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). This work will contribute to improved Arctic forecasts and the Arctic ice prediction demonstration project for the Earth System Prediction Capability (ESPC).

APPROACH

We 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 complete application codes. Our focus is on Intel many-core architectures with an emphasis on threading and vectorization improvements.

WORK COMPLETED

CICE

We explored multiple strategies to improve performance of the CICE model on the Intel Xeon Phi Knight's Landing (KNL) architecture using profiles from the Intel Vtune profiling tool as a guide. We have been benchmarking CICE on a local LANL platform (Trinitite) that consists of nodes with 68-core KNL chips paired with a Haswell CPU. The 68 cores on the KNL are laid out in a 2D mesh and local high-bandwidth memory can be assigned to four quadrants of that mesh as additional addressable memory (quad, flat) or as another level of cache (quad, cache). Each KNL core has 4 hardware threads available. The results below refer to two benchmark problems, called tp1 (a 0.1-degree resolution) and p4 (0.4-degree resolution), with OpenMP threading enabled. Table 1 below shows the initial timings of running the p4 problem for 48 simulation hours on one KNL node configured as quad flat.

# MPI Ranks x # Threads	Time (secs)
Baseline SVN revision r1158 64 ranks	202
64x1	210
32x2	203
32x4	175
16x4	193
16x8	213
8x8	329

Table 1. Timings of p4 problem for 48 simulated hours on KNL with OpenMP threading

The table shows that the fastest time was running with 32 MPI ranks, each rank using 4 threads. In this mode, we are using 2 hardware threads per core. If we keep the total number of processing units constant, e.g. 64 PUs, we see a minor improvement in thread scaling compared to the Haswell performance. That is, we do see some improvement going up to 4 threads, whereas on the Haswell, the best time was for two threads. However, if we compare node-to-node of KNL to Haswell, the baseline reference run is about 2x slower on KNL than Haswell. This is to be expected. Even though the number of cores is greater, the clock speed per core is substantially lower on KNL than Haswell. To make up the performance gap in clock speed, each KNL core provides two 512-bit vector units. Therefore, to get performant code on KNL, it becomes imperative to vectorize.

Table 2 shows timings of the larger tp1 problem running on 10 KNL nodes, again using the quad flat configuration. The 80 ranks with 2 threads per rank showed slightly slower performance than the baseline. However, if we make use of the hardware threads available, then we can improve the performance by almost 2x. This is crucial as memory pressure prohibits running this problem with more than 16 ranks on a KNL node.

# MPI Ranks x # Threads	Time (secs)
Baseline SVN revision r1158 160 ranks	2764
80x2	2795
80x4	1514
80x8	1512

Table 2. Timings of tp1 problem on KNL with OpenMP threading

Much of our focus this year was improving vectorization for KNL. We rely heavily on the vectorization profiling tool, Advisor, that is part of the Intel suite of performance profiling tools. Within CICE, one aspect that inhibits vectorization is the treatment of ice-covered and ice-free regions, using indirect addressing to gather active ice regions. Although the chip hardware provides support for gather/scatter, there is still a cost when a potentially vectorizable loop contains indirect addressing. We implemented a couple of solutions. One approach we implemented for vectorization was to use temporary arrays to avoid random array access and place the data used in contiguous memory. We did achieve better vectorization, but at the cost of more memory. For other loops, we added directives to force vectorization where profiling results claimed falsely about the existence of data dependencies.

Our second round of benchmarking in Table 3 below shows that KNL performance is still lagging the Haswell architecture significantly. While somewhat slower performance is typical of first implementations on KNL, the baseline performance on 10 KNL nodes versus 10 Haswell nodes is more than three times slower. We had expected that running with fewer ranks, but with threads, would be faster than running MPI everywhere. When running MPI everywhere, our performance profiles showed a large amount of time was spent in MPI during halo exchanges, implying load imbalance. The rationale was with fewer ranks, more cells would contain sea ice and increase the amount of computation for each rank, and introducing threads would fill the computing gap. The KNL timings support our conjecture, but the Haswell results are contradictory. However, we will continue to focus our efforts on KNL and explore methods to get the timings closer to Haswell performance.

# of ranks x # of threads/rank	Solution Time (Seconds): Haswell	Solution Time (Seconds): KNL
Baseline 320x1	433	1505
160x2	477	No data
80x4	502	1396

Table 3. Solution times for tp1 integrated 48 simulation hours on 10 nodes

As noted above, the most significant performance issue with CICE is related to communication and load balancing. This appears as another profile hotspot in the *MPI_Waitall* function occurring within halo updates. This halo update call is performed frequently due to subcycling of the fast waves within the sea-ice dynamics routine. The halo exchange is serialized and follows a parallel region where stresses are computed. However, we can't perform the halo update in a separate thread because the data depends on the previous loop where the velocities are updated. We therefore attempted to parallelize the halo update routine itself. We wrote an asynchronous version that replaces the *MPI_Waitall* with a loop using

MPI_Test on the receive requests, and we put the loop inside an OpenMP threaded region. Unfortunately, this approach did not work, apparently due to a non-thread-safe implementation of *MPI_Test* in the Cray MPI implementation on Trinitite. We may try a different MPI implementation to see if this strategy will work.

The complexity of these new architectures continues to cause unexpected problems when configuring resource managers for optimal performance. For example, a change to the resource manager for the Trinitite cluster caused the tp1 problem with 80 ranks and four threads per rank across six nodes to fail, despite previous successful runs. We spent significant time attempting to resolve the source of the issue. It was finally determined that we had to add an additional argument, '--vm-overcommit=enable', when launching through the resource manager. With this fix, we were able to run this configuration again. In a second instance, the Vtune Memory Analysis showed a high rate of memory access on a different Non-Uniform Memory Access (NUMA) domain when running on a Haswell node. A Haswell node on Trinitite consists of two sockets and each socket holds one NUMA domain. Memory access to a non-local NUMA domain incurs larger latency. Even though we were binding all threads of an MPI rank to the same socket, memory was still being accessed from the other socket. Eventually, it was determined that we were missing an option for *srun*, the job launcher on Trinitite that explicitly instructs the launcher to bind processes to NUMA domains. Using this option almost eliminated entirely cross-NUMA memory access. Compiler optimizations also play a role. We found a nontrivial amount of time was spent in a call to *__powr8i4*. Examining the callstack, we traced it to taking exponents larger than 2, e.g. *x**3*. Transforming this by expanding to *x*x*x* greatly reduced the time of *__powr8i4*.

Finally, we updated both the baseline code and current code to revision 1232 in the zbgc_colpkg branch of the main CICE repository. The latest benchmarks are show in Tables 4-6 below that include all the improvements described above. Our modifications give slight improvements on the Haswell CPU, but more significant improvements (up to 15%) on the KNL architecture. This reflects mostly our efforts with improving vectorization, which is critical for KNL performance. However, KNL performance continues to lag significantly behind Haswell.

# of ranks x # of threads/rank	Baseline	New Code
320x1	423	415
160x2	418	406
80x4	415	410

Table 4. Haswell runs of tp1 for 48 simulation hours on 10 nodes (time in seconds)

# of ranks x # of threads/rank	Baseline	New Code
320x1	1273	1117
160x2	1279	1082
80x4	1250	1105

Table 5. KNL runs of tp1 for 48 simulation hours on 10 nodes using quad-cache mode (seconds)

# of ranks x # of threads/rank	Baseline	New Code
320x1	1347	1270
160x2	1322	1242
80x4	1274	1204

Table 6. KNL runs of tp1 for 48 simulation hours on 10 nodes using quad-flat mode (seconds)

We presented our work on CICE during the KNL Application Experience Workshop held at Sandia National Laboratory (Rob Aulwes was the primary organizer of this workshop). The workshop included participants from Berkeley Lab (NERSC), Lawrence Livermore Lab, and Sandia Lab. The purpose of the workshop was to provide a venue for applications to exchange ideas and experiences with adapting their codes for the KNL architecture. The workshop was held from July 11 – 13, 2017.

HYCOM

The Knights Landing nodes on the new Cray XC40 system at ERDC DSRC were not scheduled to be available until July 5th. While waiting for availability, we set up a small global HYCOM (0.72 degree tripole) test case designed to fit on a single 64-core Knights Landing node. Figure 1 shows the 64 MPI-task layout which discards 6 all-land tiles from a 10x7 decomposition (active tile size 50x55). This is primarily for testing the performance of the re-factored KPP mixed layer sub-model, but can also be used to tune the rest of the model for KNL. This configuration will permit testing to begin on a small KNL test system at ARL DSRC. HPCMP PETT has started profiling HYCOM on that system using test cases provided by Alan Wallcraft.

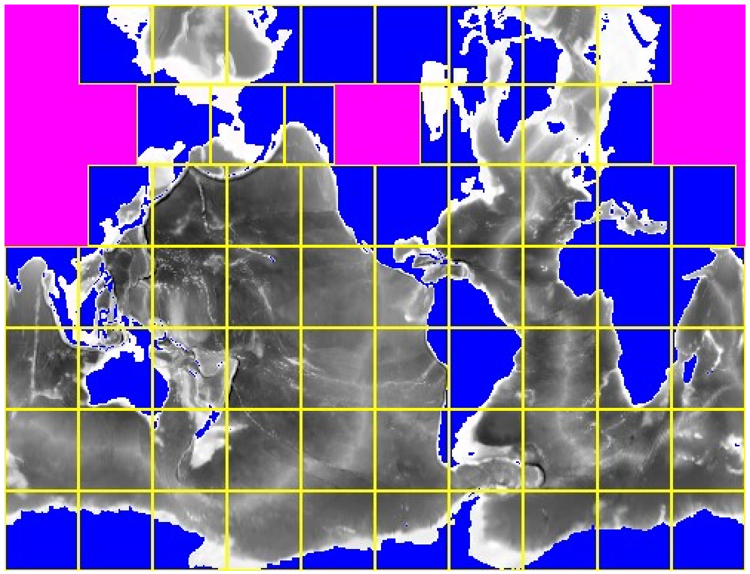
Figure 1. Domain decomposition for 0.72 tripole test case

The Cray XC40 system at ERDC DSRC is now available to start our tuning of HYCOM for this processor. It will have 34K Knights Landing cores (540 64-core nodes), in addition to 126K Broadwell cores. This system is large enough to handle our 1/25 degree HYCOM benchmark case. Alan presented APPIGO results at the HPCMP CWO Workshop on 22 February, and at NRL's 6.2/6.4 reviews on 22 March. Alan's affiliation has also changed this year, moving from the Naval Research Lab to join Eric Chassignet at Florida State.

HYCOM in CESM

We continue the evaluation of baseline experiments for validation of model improvements using both a version of the Community Earth System Model (CESM) configured with HYCOM and CICE and an ESPC coupled system with NAVGEM. The CESM-HYCOM coupled system (ocean-sea ice-atmosphere-land-runoff) is running on both a 1/12^o HYCOM grid (atmospheric grid at 0.47^{o}x0.63^o) and at 0.72^o (atmospheric grid at 1.9^{o}x2.5^o). In parallel, the ESPC coupled system with NAVGEM as its atmospheric component and HYCOM as its ocean component was evaluated on Gordon (NAVY, Cray XC40) with the same HYCOM grids at 0.72^o and 1/12^o resolutions than the CESM system.

Results for the CESM-HYCOM 1/12^o configuration after 10 years of integration show a decrease of the biases at the surface for the sea surface temperature and globally in terms of temperature. The global salinity shows a similar behavior at both resolutions with a very small positive drift. CESM-HYCOM 1/12^o also shows similar surface temperature biases as the CESM-POP 1/10^o except in the North Atlantic sub-polar gyre (stronger cold bias in CESM-POP) and equatorial Pacific (warm bias in CESM-POP) (see McClean et al. 2011). The analyses of the surface fluxes, streamfunctions and transports at different sections also revealed an improvement at high resolution with, for some diagnostics, better results with HYCOM than with default POP ocean model. The simulation CESM-HYCOM 1/12º is now being extended to 20 years. CESM simulations are performed on Yellowstone (UCAR, IBM iDataPlex). The high resolution uses 2000 processors and runs a day in ~12min, the low resolution uses 80 processors and runs 5 days in under 2 min.

The ESPC coupled system with NAVGEM as its atmospheric component and HYCOM as its ocean component was evaluated on Gordon (NAVY, Cray XC40) with the same HYCOM grids at 0.72º and 1/12º resolutions than the CESM system. Both resolutions were run for five years. As in CESM, the global drift in temperature decreases in the higher resolution simulation, to reach almost zero. Unlike in CESM, the sea surface temperature bias is not decreasing with resolution but mostly reverse. While the SST has a positive drift at 0.72º, the SST decreases over time at 1/12º. Transports and streamfunction are similar at both resolutions. The ESPC-HYCOM 1/12º has been run on 900 procs (ocn/ice) + 96 procs (atm T319) and produced a daily archive in 25-26 min, while the ESPC-HYCOM 0.72º runs on 40 procs (ocn+ice) + 96 procs (atm T119) and produced a 5-day archive in ~5min. The evaluation of the ESPC-HYCOM coupled with NAVGEM on a 1/12º and 0.72º grid also shows a gradual reduction of the sea-ice cover in the Antarctic at both resolutions (Fig. 2 for high-resolution). When reviewing the exchanged fields between CICE and NAVGEM, we found a bug in the way the units of precipitations received by CICE making it 1000 times lower than it should be.

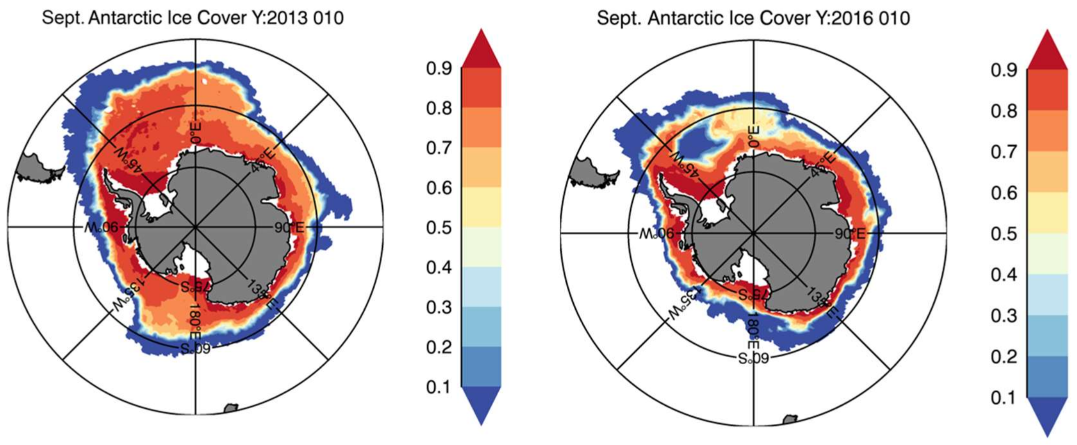
Sept. Antarctic Ice Cover Y:2013 010
Sept. Antarctic Ice Cover Y:2016 010
0°E
45°E
90°E
135°E
180°E
135°W
90°W
45°W
60°S
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Fig. 2: Ice cover in Austral winter for year 2 (left) and year 5 (right) for ESPC-HYCOM 1/12º global.

Tests were therefore performed with corrected precipitations in CICE at both resolutions and using the latest available version of NAVGEM with the CV2 physics. Results show now a more stable ice extent in the Antarctic for the low resolution (T119-GLBt0.72) when compared with the older simulation (red lines Fig. 3), but also a gradual accumulation of ice over the years in the Arctic due to a lower melting rate during the summer.

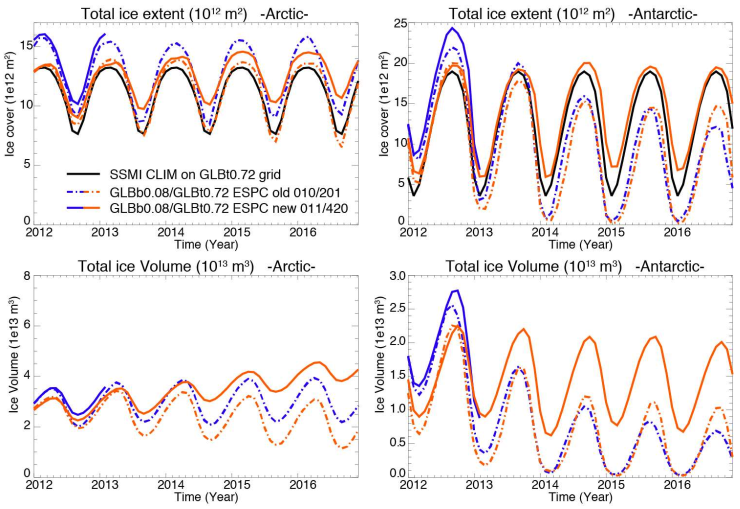

Fig. 3: Time series of ice extent (top) and volume (bottom) in the Arctic (left) and Antarctic (right). In black is the NSIDC/SSMI ice extent climatology, in blue the results for the 1/12º and in red for the 0.72º. Plain/dashed lines are for the corrected/non-corrected.

The high-resolution simulation (T359-GLBb0.08) has run only 14 months for now but already presents a higher ice extent in the Antarctic region than with the older simulation (blue lines in Fig. 3). As in the low resolution, less melting occurs in both polar regions during their respective summer, but it is still unclear if the ice volume will continue to rise in the Arctic during the simulation as is seen in the low resolution. More information will be available once the 5-year simulation is complete.

In order to separate the impact of the new NAVGEM version from the CICE bug fix, three low resolutions simulations were performed using the different versions of NAVGEM available in the ESPC system: 1-NAVGEM-CV1.01, 2- NAVGEM-v1.3-rc2 and 3-NAVGEM-CV2. While all experiments have a better Antarctic ice extent and volume than without the bug fix, all three presents an increase of the ice extent and volume in the Arctic (Fig. 4).

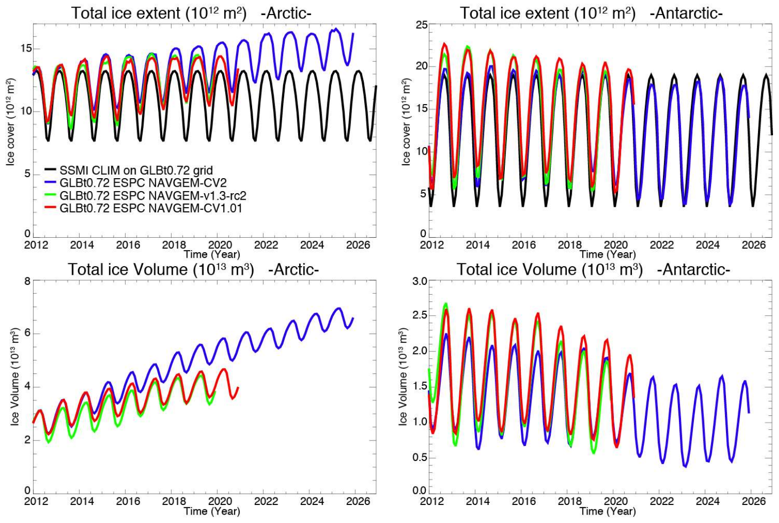

Fig. 4: Time series of ice extent (top) and volume (bottom) in the Arctic (left) and Antarctic (right). In black is the NSIDC/SSMI ice extent climatology, in blue the results for the NAVGEM-CV2, in green for NAVGEM-v1.3-rc2 and in red for NAVGEM-CV1.01.

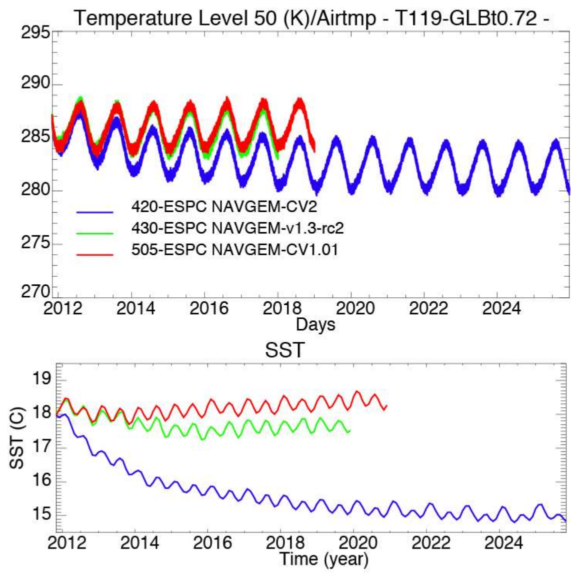

Fig. 5: Time series of the global average air temperature (top) and SST (bottom). In blue the results for the NAVGEM-CV2, in green for NAVGEM-v1.3-rc2 and in red for NAVGEM-CV1.01.

While NAVGEM-CV1.01 and NAVGEM-v1.3-rc2 have a similar increase rate in the Arctic, NAVGEM-CV2 ice volume increases faster than in the other two simulations, and is not stabilized after 14 years of simulation. One explanation for this behavior might be the decreasing surface temperature in NAVGEM-CV2 leading to a significant decrease of the SST from a global average mean of 18ºC to 15ºC (Fig. 5).

Despite the choice of new convective parameters for the low resolution, the NAVGEM-CV2 source code is still behaving unrealistically on a low-resolution grid (T119) and identical diagnostics are going to be performed on the high resolution to see if the same problems occur.

RESULTS

We have obtained early benchmark results on new KNL systems and achieved some moderate (15%) improvements.

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.