Porting CESM+MOM6 Ocean Models to Multiple Architectures

G. Dylan Dickerson
University of Wyoming

Supreeth Suresh
NCAR, Mentor

July 31, 2019
Contents

- Project Goals
- Background
- Challenges
- Development Cycle
- Results
- Lessons Learned
- Future Work
Project Goals

- Port CESM2+MOM6 onto Graphical Processing Units (GPUs)
  - Portability
  - Reasonable performance gains
  - Minimal Code Change
- Train me to continue University of Wyoming and NCAR collaboration on this and other projects
- Enhance the portability of CESM
# How accurate is your forecast?

<table>
<thead>
<tr>
<th></th>
<th>90% accurate</th>
<th></th>
<th>80% accurate</th>
<th></th>
<th>50% accurate</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>☀️ 76°</td>
<td></td>
<td>☀️ 80%</td>
<td></td>
<td>☞??</td>
<td></td>
</tr>
<tr>
<td>Tue</td>
<td>☁️ 74°</td>
<td></td>
<td>☁️ 80%</td>
<td></td>
<td>☞??</td>
<td></td>
</tr>
<tr>
<td>Wed</td>
<td>⛈ 70°</td>
<td></td>
<td>☂️ 80%</td>
<td></td>
<td>☞??</td>
<td></td>
</tr>
<tr>
<td>Thu</td>
<td>☔️ 70°</td>
<td></td>
<td>☔️ 80%</td>
<td></td>
<td>☛??</td>
<td></td>
</tr>
<tr>
<td>Fri</td>
<td>☁️ 71°</td>
<td></td>
<td>☁️ 80%</td>
<td></td>
<td>☛??</td>
<td></td>
</tr>
<tr>
<td>Sat</td>
<td>☀️ 76°</td>
<td></td>
<td>☀️ 80%</td>
<td></td>
<td>☞??</td>
<td></td>
</tr>
<tr>
<td>Sun</td>
<td>☀️ 75°</td>
<td></td>
<td>☀️ 80%</td>
<td></td>
<td>☞??</td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>☞??</td>
<td></td>
<td>☞??</td>
<td></td>
<td>☛??</td>
<td></td>
</tr>
<tr>
<td>Tue</td>
<td>☞??</td>
<td></td>
<td>☛??</td>
<td></td>
<td>☛??</td>
<td></td>
</tr>
<tr>
<td>Wed</td>
<td>☞??</td>
<td></td>
<td>☛??</td>
<td></td>
<td>☛??</td>
<td></td>
</tr>
</tbody>
</table>
What about seasonal models/forecasts?

Snowboarder at Powder Mountain

Campsite on Casper Mountain

Harvesting on a farm in Kansas
Parts of CESM

- Atmosphere
- Land
- Land-Ice
- Waves
- River Runoff
- Ocean
- Sea-Ice

Coupler
Short Name: CMOM

Long Name: 2000_DATM%NYF_SLND_DICE%SSMI_MOM6_DROF%NYF_SGLC_SWAV

Features of MOM6
• Highly scalable
• Open source
• Robust
Challenges

• Learning Fortran and MPI
  – Find $Z = aX + Y$ and then use a matrix library to find inverse in Fortran90
  – Build utilizing Makefiles
  – Decomposing work among MPI-tasks
  – Add OpenACC directives to utilize GPUs
• Submit and run jobs on Casper and Cheyenne

Fortran Code: https://github.com/gdicker1/MPI_practice
Challenges

• Working with CESM, CIME, and MOM6
  – How to build and execute CESM
  – How to change configuration parameters in CIME
    • Adding Casper system to machine list
    • Configuring for PGI compiler on Casper
    • Updating PGI compiler configuration for Cheyenne
  – Configuring a case
    • Changing run parameters, especially required number of tasks
Challenges

• Non-uniform software stacks
• Intel and PGI interpret standards differently
• HPC upgrades and outages
• Code modifications on GitHub
• Profiling information from MOM6
Development Cycle

- Successfully build
- Successfully execute
- Validate
- Profile code
- Analyze code
- Future Work
Profiling System Information

Test case: CMOM compset
Using NCAR Cheyenne Supercomputer
- 2x 18-core Intel Xeon version 4 (Broadwell)
- PGI compiler version 19.3
  - MPI Library: OpenMPI version 3.1.4
- Intel compiler version 17.0.1
  - MPI Library: MPT version 2.16
- 1x EDR IB interconnect

Runs on Casper Supercomputer
- 2x 18-core Intel Xeon Gold 6140 (Skylake)
- PGI compiler version 19.4
  - MPI Library: OpenMPI version 3.1.4
Porting CESM+MOM6

Profiling Results

MOM WITH CMOM COMPSET

- Ocean Dynamics: 57%
- Ocean Thermodynamics & Tracers: 27%
- Other: 16%

Built with Intel compilers, 144 tasks, 4 compute nodes, 5 simulated days
Profiling Results

OCEAN DYNAMICS

- Message Passing: 3%
- Vertical Viscosity: 11%
- Continuity: 12%
- Pressure Force: 11%
- Horizontal Viscosity: 5%
- Coriolis & MOM Advection: 3%
- ALE: 6%
- Diffusion via CVMix: 1%
- Barotropic Stepping: 16%
- Other: 31%
- Barotropic Forcing Calculation: 1%

Built with Intel compilers, 144 tasks, 4 compute nodes, 5 simulated days
## Results

<table>
<thead>
<tr>
<th>Name</th>
<th>Average Time (secs)</th>
<th>Number of Source Code Lines</th>
<th>Call Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertical Viscosity</td>
<td>25.2</td>
<td>876</td>
<td>2</td>
</tr>
<tr>
<td>Continuity Equation</td>
<td>38.8</td>
<td>1396</td>
<td>2</td>
</tr>
<tr>
<td>Pressure Force</td>
<td>23.7</td>
<td>842</td>
<td>3</td>
</tr>
<tr>
<td>Barotropic Step</td>
<td>35.4</td>
<td>1880</td>
<td>1</td>
</tr>
</tbody>
</table>

Built with Intel compilers, 144 tasks, 4 compute nodes, 5 simulated days
Results

Comparison of PGI & Intel Runs

- Total runtime: 2.13x
- Ocean: 2.17x
- Ocean dynamics: 1.92x

144 tasks, 4 compute nodes, 5 simulated days
Future Work

- Train other students at University of Wyoming
- Correct results with PGI compiler
- Gather more profiling data
  - Especially calls per timestep
- Port and parallelize routines in Ocean Dynamics
- Extract parallelism across multiple GPUs
- Prepare poster for Supercomputing 2019
Lessons Learned

- Knowledge on programming in Fortran90 and parallelization with MPI
- Ported CESM to a new architecture
- Compiler interpret standards differently
- How to communicate of compiler errors
- Difficulty of applying profiling or debugging tools with a large project
Acknowledgements

• Dr. Raghu Raj Kumar – NVIDIA
• Brian Dobbins – NCAR, CISL
• Dr. Gustavo Marques – NCAR, CGD
• Dr. Michael Levy – NCAR, CGD
• Dr. Carl Ponder – NVIDIA
• Dr. Richard Loft – NCAR, CISL
• Henry O’Meara – University of Wyoming, NCAR
• AJ Lauer – NCAR, CODE
• Virginia Do – NCAR, CODE
• Eliott Foust – NCAR, CODE
• Ingrid Jo, Clint Walker, and Samantha Williams for their pictures
All logo images were source from each organization’s branding website

2. Ingrid Jo for camping picture
3. Clint Walker for snowboarding picture
4. Samantha Williams for farming picture
8. ”Resources.” Supercomputing https://sc19.supercomputing.org/attend/media/resources/#section1
G. Dylan Dickerson
geo.dickerson21@outlook.com
github.com/gdicker1
Thank You for Attending