CESM Load Balancing Development: Python Scripts for Workflow

Thomas Johnson III (Elizabeth City State University)
and Soudeh Kamali (University of Wyoming)
+Sheri Mickelson, Brian Dobbins, John Dennis

July 30th, 2020
What is Load Balancing?

- Load balancing is a process of actively managing resources.
- Applied by determining which tasks should receive a given amount of finite resources.
What is CESM?

- Stands for Community Earth System Model.
- Software for simulating the weather and climate systems of the Earth.
- Critical tool for climate studies.
Why Load Balancing for CESM?

- Building and running CESM models can be resource expensive.
- Provide guidance for running CESM for newcomers.
- Enable CESM to be run across multiple environments effectively.
Current Status for CESM

- Manual application of load balancing and running CESM models.
- Users must guess the best scheme for allocating resources.
Resources Utilized

- Software components: Python, Bash, CESM
- Hardware Components: Cheyenne
• Specify the max number of tasks in the command line.
• Specify the number of CESM model runs.
Core Allocation

- Cores are allocated for each component, with WAV capped to preserve efficiency.
- CESM models are built and run.
Load Balancer Called

- User decides to run load balancing software on outputted timing files.
- Load balancing software processes the timing files.
Load Balancer Process

- Load balancing software produces optimized values based on provided timing files.
- Said optimized values are stored into a JSON file.
- JSON files can be turned into a python dictionary to be processed by CESM.
Automated Load Balancing Continued

- The user now has optimized values to build and run future CESM models with.
- Start to finish in one process.
Command Line Input

• An Example of Running the CESM Load Balancing Code:
  – % python cesm_allocation.py 288 --compset_designation B1850 --sim_time_designation 2 --sim_time_unit ndays
B1850 2 degree for ATM and 1 degree for OCN run Statistics.

<table>
<thead>
<tr>
<th>Number of Days Simulated</th>
<th>2 Days</th>
<th>5 Days</th>
<th>7 Days</th>
<th>12 Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Difference of Seconds Per Model Day to 2 Simulated Days</td>
<td>0%</td>
<td>0.61%</td>
<td>0.70%</td>
<td>0.78%</td>
</tr>
<tr>
<td>Absolute Seconds Per Model Day</td>
<td>40.43</td>
<td>40.18</td>
<td>40.15</td>
<td>40.12</td>
</tr>
</tbody>
</table>
OCN Component Relative Seconds Per Model Day Statistics

- B1850 2 degree for ATM and 1 degree for OCN run Statistics.

<table>
<thead>
<tr>
<th>Number of Days Simulated</th>
<th>2 Days</th>
<th>5 Days</th>
<th>7 Days</th>
<th>12 Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative Difference of Seconds Per Model Day to 2 Simulated Days</td>
<td>0%</td>
<td>3.9%</td>
<td>4.5%</td>
<td>5.7%</td>
</tr>
<tr>
<td>Absolute Seconds Per Model Day</td>
<td>55.06</td>
<td>52.91</td>
<td>52.59</td>
<td>51.90</td>
</tr>
</tbody>
</table>
• B1850 1 degree for ATM and 1 degree for OCN run Statistics.

<table>
<thead>
<tr>
<th>Time Statistics of Runs On 5 Day Simulation Settings of ATM Component</th>
<th>Time Statistics of Runs On 5 Day Simulation Settings of OCN Component</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average Seconds Per Model Day</strong></td>
<td><strong>Average Seconds Per Model Day</strong></td>
</tr>
<tr>
<td>178.41</td>
<td>52.93</td>
</tr>
<tr>
<td><strong>Maximum Seconds Per Model Day</strong></td>
<td><strong>Maximum Seconds Per Model Day</strong></td>
</tr>
<tr>
<td>178.50</td>
<td>53.23</td>
</tr>
<tr>
<td><strong>Minimum Seconds Per Model Day</strong></td>
<td><strong>Minimum Seconds Per Model Day</strong></td>
</tr>
<tr>
<td>178.34</td>
<td>52.60</td>
</tr>
</tbody>
</table>
Implemented Features

- Build CESM models and utilize load balancer in one process.
- Ability to select compsets.
- Automatically scales out components.
Added Features

- JSON files for storing and loading CESM parameters.
- Restriction for the WAV component to prevent inefficiency from excess core usage.
- Specify the number of CESM models to be built and ran concurrently.
Future Work

• Implementing more options for scaling.
• Exploring more options for configuration of building CESM models.
Acknowledgements

• Thanks to the SIParCS Program and NCAR for providing this opportunity.
• Thanks to Soudeh Kamali for being an excellent SIParCS partner and supportive colleague and extracting data as well as acquiring summary statistics.
• Thanks to Sheri Mickelson, Brian Dobbins, and John Dennis for being incredible SIParCS mentors offering guidance and knowledge.
• Load Balancing Software by Jim Edwards et al. DOI:http://dx.doi.org/10.5065/WE0D-9K91. URL: https://github.com/ESMCI/cime.
• Image citations within the alt text.
Questions

- Thomas Hilton Johnson III
- Email: Thomash.johnson261@gmail.com