Performance Analysis of DroughtHPC and Holistic HPC Workflows

Yasodha Suriyakumar
Karen L. Karavanic
Hamid Moradkhani

yasodhan@pdx.edu
karavan@pdx.edu
hmoradkhani@ua.edu

Goals

• Analyze the performance of DroughtHPC [1], a drought prediction application developed at Portland State University
  o DroughtHPC improves prediction accuracy for a target geographical area; uses data assimilation techniques that integrate data from hydrologic models, and satellite data
  o Uses variety of data: soil conditions, snow accumulation, vegetation layers, canopy cover and meteorological data
• Scale the application to do finer-grained simulations, and to simulate a larger geographical area

Implementation

• Land surface of the target geographical area is modelled as grid of uniform cells, and simulation divides it into jobs, with group of 25 cells in each job
• For a job that simulates 50 meteorological samples and one month time period, input data size is 144.5 Mb, with the satellite data consuming 132 Mb
• Runtime for a job on single-node is approximately two hours with the initial Python prototype

Results: Single Node

Runtime data of DroughtHPC (with VIC) for 50 meteorological samples and one month simulation on a single node (8 cores)

<table>
<thead>
<tr>
<th>Number of jobs (group of 25 cells)</th>
<th>Minimum runtime (hours: minutes)</th>
<th>Maximum runtime (hours: minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2:21</td>
<td>2:21</td>
</tr>
<tr>
<td>2</td>
<td>1:54</td>
<td>1:55</td>
</tr>
<tr>
<td>4</td>
<td>1:49</td>
<td>1:50</td>
</tr>
<tr>
<td>8</td>
<td>1:48</td>
<td>1:50</td>
</tr>
<tr>
<td>12</td>
<td>2:42</td>
<td>2:44</td>
</tr>
</tbody>
</table>

Observations

• No single performance tool characterizes the calling pattern / interactions between Python and VIC
• Profiling does not focus attention to the number of files or frequency of file creation/access
• Manual integration of data from multiple performance tools is time-consuming: Python cProfile, Valgrind, Strace system utility, etc.

Methodology

Evaluate sequential single-cell simulation performance
Analyze timing and memory footprint of hydrology models

For parallel single-node performance, study the correlation between runtime and the problem size

For parallel multi-node performance in a Linux cluster, analyze effects of interference from other processes

Summary and Future Work

• Single-cell simulations: bottleneck is the overhead of the VIC hydrologic model call from Python
• Parallel single-node performance:
  o The best fit on our platform is one job per logical core;
  o We explored changes to VIC for Intel Xeon Phi
  o We are developing a version of VIC that eases integration with individual science codes such as data assimilation
• Multi-node simulations with MPI: bottleneck is the filesystem access pattern
• We designed PPerfG for visualizing Holistic HPC Workflows
• We implemented a prototype of PPerfG

PPerfG

• PPerfG: A Visualization Tool for Holistic HPC Workflows for use in performance diagnosis
• Captures the data movement behavior between storage layers, and between different stages of an application
• Challenges: Determining best metrics, and efficient measurement techniques
• Status: initial prototype developed

References


Acknowledgements

Henry Conney, Tu Lo, and Jiaqi Luo contributed ideas, discussed implementation, and implemented software used in this project. This material is based upon work supported by the National Science Foundation Grant No. 1359065. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.