Towards Performance Portability for Atmospheric and Climate Models with the GGDM model DSL

Nabeeh Jum'ah, Julian Kunkel, Günther Zängl, Hisashi Yashiro, Thomas Dubos, Yann Meurdesoif

Scientific Computing
Department of Informatics
University of Hamburg

ISC HPC Research Poster Presentation
2017-06-20

This work was supported in part by the German Research Foundation (DFG) through the Priority Programme 1648 “Software for Exascale Computing” (SPPEXA) (GZ: LU 1353/11-1)
Towards Higher-Level Code Design

Goals

- Improve code quality and scientists productivity
  - Enhance source repositories maintainability
  - Reduce complexity in optimized-code development
- Provide better performance-portability of code

Constraints

- The existing codebases should be preserved
- Tools should be lightweight, flexible, and easily maintainable
Improving Code Quality and Performance-Portability

Strategy

- Foster separation of concerns
  - Domain scientists develop domain logic in source code
  - Scientific programmers write hardware configurations
- Allow coding domain logic with a Domain-Specific Language
  - Extending an existing GPL with domain science concepts
  - Free of any lower level (e.g., architecture) details
- Provide the tools to implement S2S translation
  - Guided by configurations provided by scientific programmers
GGDML: Our Developed DSL

- **GGDML:** General grid definition and manipulation language
- Development: Co-design in collaboration with domain scientists
- Features
  - Hides memory access details
  - Abstracts higher concepts of grids, hiding connectivity details
- Constructs for the abstraction of grids
  - Grid definition
  - Grid-bound variable declaration
  - Grid-bound variable access/update
  - Stencil operations

![Diagram showing triangular and hexagonal grids](image-url)
GGDML Impact on Code Quality

Evaluation

We estimated changes on code size and complexity

<table>
<thead>
<tr>
<th>Model, kernel</th>
<th>lines (LOC)</th>
<th>words</th>
<th>characters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>before DSL</td>
<td>with DSL</td>
<td>before DSL</td>
</tr>
<tr>
<td>ICON 1</td>
<td>13</td>
<td>7</td>
<td>238</td>
</tr>
<tr>
<td>ICON 2</td>
<td>53</td>
<td>24</td>
<td>163</td>
</tr>
<tr>
<td>NICAM 1</td>
<td>7</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>NICAM 2</td>
<td>90</td>
<td>11</td>
<td>344</td>
</tr>
<tr>
<td>DYNAMICO 1</td>
<td>7</td>
<td>4</td>
<td>96</td>
</tr>
<tr>
<td>DYNAMICO 2</td>
<td>13</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>total</td>
<td>183</td>
<td>55</td>
<td>911</td>
</tr>
<tr>
<td>percentage</td>
<td>30.05%</td>
<td>47.20%</td>
<td>45.04%</td>
</tr>
</tbody>
</table>

We investigated potential cost savings using COCOMO

<table>
<thead>
<tr>
<th>Software project</th>
<th>DSL?</th>
<th>Effort Applied</th>
<th>Dev. Time (months)</th>
<th>People require</th>
<th>dev. costs (M€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-detached</td>
<td>without</td>
<td>2462</td>
<td>38.5</td>
<td>64</td>
<td>12.3</td>
</tr>
<tr>
<td></td>
<td>with</td>
<td>1133</td>
<td>29.3</td>
<td>39</td>
<td>5.7</td>
</tr>
<tr>
<td>Organic</td>
<td>without</td>
<td>1295</td>
<td>38.1</td>
<td>34</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>with</td>
<td>625</td>
<td>28.9</td>
<td>22</td>
<td>3.1</td>
</tr>
</tbody>
</table>