A New OpenModelica Compiler High Performance Frontend

Adrian Pop1 Per Östlund1 Francesco Casella2 Martin Sjölund1 Rüdiger Franke3

1PELAB - Programming Environments Lab, Dept. of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden, {adrian.pop,per.ostlund,martin.sjolund}@liu.se
2Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy, francesco.casella@polimi.it
3ABB, IAPG-A26, Kallstader Str. 1, 68309 Mannheim, Germany, ruediger franke@de.abb.com

Abstract

The equation-based object-oriented Modelica language allows easy composition of models from components. It is very easy to create very large parametrized models using component arrays of models. Current open-source and commercial Modelica tools can with ease handle models with a hundred thousand equations and a thousand states. However, when the system size goes above half a million (or more) equations the tools begin to have problems with scalability. This paper presents the new frontend of the OpenModelica compiler, designed with scalability in mind. The new OpenModelica frontend can handle much larger systems than the current one with better time and memory performance. The new frontend was validated against large models from the ScalableTestSuite library and Modelica Standard Library, with good results.

The results for selected ScalableTestSuite (STS) models and the Vectorized.SolarSystem are presented in Table 1. One can see that the new OpenModelica frontend performs very well in comparison to Dymola, in some cases faster, in some cases slower. The comparison between the current frontend (CF) and the new frontend (NF) is also included where possible. From these benchmarks one can also see that investigation is needed to find out why parameter arrays are scaling poorly in the new frontend (models 6, 7, 8). For models 10 and 11 the figure in the parentheses is for the new frontend not expanding arrays at all during the flattening. The performance improvement in this case is extreme.

In Table 2 we compare the current frontend (CF) with the new frontend (NF) when instantiating and flattening models from Modelica.Mechanics.MultiBody and evaluating their graphical annotation. The OpenModelica compiler API function that is called to evaluate the graphical annotations is getComponentAnnotations(). The new frontend performs 20 to 200 times better than the current OpenModelica frontend, allowing to obtain a nearly immediate response time of the OMEdit GUI, which relies on this API.

Keywords: OpenModelica, compiler, flattening, frontend, modelling, simulation, equation-based, scalability

<table>
<thead>
<tr>
<th>No</th>
<th>Model</th>
<th>Equations</th>
<th>CF (s)</th>
<th>NF (s)</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Electrical.DSystemAC.SE.DistributionSystemLinear_N_40_M_40</td>
<td>99776</td>
<td>15.53</td>
<td>06.32</td>
<td>91.33</td>
</tr>
<tr>
<td>2</td>
<td>Electrical.DSystemAC.SE.DistributionSystemLinear_N_80_M_80</td>
<td>397936</td>
<td>40.30</td>
<td>17.76</td>
<td>231.06</td>
</tr>
<tr>
<td>3</td>
<td>Electrical.DSystemAC.SE.DistributionSystemLinear_N_112_M_112</td>
<td>779312</td>
<td>74.21</td>
<td>32.31</td>
<td>232.99</td>
</tr>
<tr>
<td>4</td>
<td>Electrical.DSystemDC.SE.DistributionSystemModelicaActiveLoads_N_80_M_80</td>
<td>129929</td>
<td>18.04</td>
<td>08.33</td>
<td>217.21</td>
</tr>
<tr>
<td>5</td>
<td>Electrical.TransmissionLine.SE.TransmissionLineModelica_N_1280</td>
<td>26915</td>
<td>17.29</td>
<td>03.99</td>
<td>43.51</td>
</tr>
<tr>
<td>6</td>
<td>Elementary.ParameterArrays.SE.Table_N_100_M_100</td>
<td>0</td>
<td>06.59</td>
<td>05.09</td>
<td>13.08</td>
</tr>
<tr>
<td>7</td>
<td>Elementary.ParameterArrays.SE.Table_N_400_M_400</td>
<td>0</td>
<td>10.25</td>
<td>12.19</td>
<td>0.85</td>
</tr>
<tr>
<td>8</td>
<td>Elementary.ParameterArrays.SE.Table_N_1600_M_100</td>
<td>0</td>
<td>09.77</td>
<td>19.04</td>
<td>0.51</td>
</tr>
<tr>
<td>9</td>
<td>Vectorized.SolarSystem(n=10000)</td>
<td>60001</td>
<td>146.30</td>
<td>34.12</td>
<td>4.27</td>
</tr>
<tr>
<td>10</td>
<td>Vectorized.SolarSystem(n=100000)</td>
<td>600001</td>
<td>14458.68</td>
<td>2450.57</td>
<td>6.0</td>
</tr>
</tbody>
</table>

Table 1. Flattening performance comparison Dymola vs. OpenModelica (NF vs CF included). Bold numbers in parentheses are with Scalarization disabled -d=-nfScalarize. Shortened names: SE=ScaledExperiments, DSystem=DistributionSystem.

<table>
<thead>
<tr>
<th>Model</th>
<th>CF (s)</th>
<th>NF (s)</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>World</td>
<td>9.53</td>
<td>0.28</td>
<td>33.9</td>
</tr>
<tr>
<td>Joints.FreeMotionScalarInit</td>
<td>28.90</td>
<td>0.14</td>
<td>205.0</td>
</tr>
<tr>
<td>Joints.Planar</td>
<td>3.56</td>
<td>0.13</td>
<td>27.6</td>
</tr>
<tr>
<td>Joints.UniversalSpherical</td>
<td>6.99</td>
<td>0.22</td>
<td>30.5</td>
</tr>
<tr>
<td>Joints.SphericalSpherical</td>
<td>4.64</td>
<td>0.11</td>
<td>41.4</td>
</tr>
<tr>
<td>Joints.Universal</td>
<td>2.31</td>
<td>0.12</td>
<td>19.4</td>
</tr>
</tbody>
</table>

Table 2. Flattening performance comparison of the current (old) vs the new frontend in OpenModelica (OMEdit GUI impact).