A New OpenModelica Compiler High Performance Frontend

Adrian Pop[1] Per Östlund[1] Francesco Casella[2] Martin Sjölund[1] Rüdiger Franke[3]

[1]PELAB - Programming Environments Lab, Dept. of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden, {adrian.pop,per.ostlund,martin.sjolund}@liu.se
[2]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy, francesco.casella@polimi.it
[3]ABB, IAPG-A26, Kallstadter 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 comparision 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 annoations 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

No	Model	Equations	Dym (s)	OMC NF/CF (s)
1	Electrical.DSystemAC.SE.DistributionSystemLinear_N_40_M_40	99776	15.53	06.32 / 91.33
2	Electrical.DSystemAC.SE.DistributionSystemLinear_N_80_M_80	397936	40.50	17.76 / 435.32
3	Electrical.DSystemAC.SE.DistributionSystemLinear_N_112_M_112	779312	74.21	32.31 / 1076.54
4	Electrical.DSystemDC.SE.DistributionSystemModelicaActiveLoads_N_80_M_80	129929	18.04	08.33 / 159.28
5	Electrical.TransmissionLine.SE.TransmissionLineModelica_N_1280	26915	09.84	04.45 / 47.77
6	Elementary.ParameterArrays.SE.Table_N_100_M_100	0	06.59	05.09 / 06.21
7	Elementary.ParameterArrays.SE.Table_N_400_M_400	0	10.25	12.19 / 18.03
8	Elementary.ParameterArrays.SE.Table_N_1600_M_100	0	09.77	19.04 / 28.17
9	Power.ConceptualPowerSystem.SE.PowerSystemStepLoad_N_64_M_16	11907	17.29	03.99 / 28.57
10	Vectorized.SolarSystem(n=10000)	60001	146.30	34.12 / 314.8 (**02.95**)
11	Vectorized.SolarSystem(n=100000)	600001	14458.68	2450.57 / 19760.42 (**02.95**)

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.

Model	CF (s)	NF (s)	Factor
World	9.53	0.28	33.9
Joints.FreeMotionScalarInit	28.90	0.14	199.4
Joints.Planar	3.56	0.13	25.6
Joints.UniversalSpherical	6.99	0.22	30.5
Joints.SphericalSpherical	4.64	0.11	39.5
Joints.Universal	2.31	0.12	18.4

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