A MultiBody Performance Story

Adrian Pop

2011-02-07

Open Source Modelica Consortium
Programming Environment Laboratory
Department of Computer and Information Science
Linköping University

www.OpenModelica.org
Outline

- Multi Body Engine V6 Example
- First Run
- Performance Analysis
- Action Plan (performance enhancement)
- Performance Results
- Conclusions
Once upon a time, early 2010 ...

MultiBody Engine V6

- An example in the Modelica MultiBody library

- Modelica.Mechanics.MultiBody.Examples.Loops.EngineV6 has 12491 equation(s) and 12491 variable(s). 8732 of these are trivial equation(s).
Early 2010 situation

- Flattening of EngineV6 took
  - 430 minutes, ~7.2 hours
  - Very disappointing! (Drama)
  - [the actual first run took ~2 days]

- Flattening performance had to be addressed
All tests were done on my laptop

- HP EliteBook 8440p, RAM 8Gb, HDD 160 SSD
- Intel® Core™ i7-620M Processor at 2.66 GHz
What took most of the time when flattening

- Inner-Outer handling
- Constant function evaluation via dll loading
- Handling of connections sets
- Handling of conditional components
- Handling of component reference equality
- Parsing
Inner-Outer handling

- Complete redesign
- Do not instantiate the outer at all, just use the cached instantiation of the inner (when we do not have inner outer)
- Performance impact
  - Depends on the number of outer models
  - ~65% faster

- Implemented by adrpo
Constant function evaluation via dll loading

- Do not use this at all, instead do function interpretation
- Function interpretation had to be improved as the previous implementation was rather naive
- Performance impact
  - ~40% faster

Implemented by Per Östlund
Handling of connection sets

- A redesign came up while planning the implementation of stream connectors
- Handling of default set-to-zero connections
- Performance impact
  - ~30% faster

Implemented by Per Östlund
Handling of conditional components

- A redesign came up while planning the implementation of stream connectors
- Do not instantiate conditional components at all when condition is false
- Performance impact
  - ~70% faster

Implemented by Per Östlund
Handling of component reference equality

- Better functionality for checking component reference equality
- Compare structures first then strings
- Performance impact
  - ~93% faster

- Implemented by adrpo
Parsing MSL 3.1 (everything)

- The old parser based on ANTLR2 was rather slow (~23 seconds)
- The new parser based on ANTLR3 was much better (~4 seconds)
- Performance impact
  - ~82% faster for parsing

- Implemented by Martin Sjölund & adrpo
Early 2011 situation

- Flattening of EngineV6 now takes
  - ~20 seconds
- Flattening of EngineV6 took
  - 430 minutes, ~7.2 hours

- Flattening performance impact
  - ~99.92% faster / ~1290 times faster
  - Is this a happy ending for the MultiBody story? Not yet 😊 I would say. It could be even better.
A MultiBody Story - Happy End?

- Took ~1 year and a lot of re-design and new implementation to make the flattening of MultiBody Engine V6 ~1290 times faster.

- It was less time than that, as we handled many other things during this time.
Thank You!
Questions?

OpenModelica Project
http://www.OpenModelica.org

Disclaimer
The numbers in this story are approximations and any resemblance to real data, is purely coincidental.