**Summary**

The paper presents C++ compiler toolchain for Component and connector frameworks, which are widely used in the automotive industry. In particular, the paper presents the technical features of the tool chain, as well as the evaluation of the compiler optimizations that are able to significantly decrease the execution times of generated modules compared to similar modules written using different means.

The artifact consists in a companion page containing different facets of the work, including, a web-based version of the EmbeddedMontiArc solution, video tutorials, as well as materials and source code used for the case study.

In its current shape, the artifact contains a lot of material, and overall, it could be considered between « functional » and « reusable », with some restrictions due to MathLab simulink license issue and restrictive platform requirements. Moving the current repository to an independent (REMODD-like) repository could contribute to increase its score to an additional artifact « available ». My main concern is related to the flow of the generator part, where all the elements could be better articulated through a slightly more elaborated description of the second part of the artifact. This would provide a reader a comprehensive global picture.

**Consistent with the paper**

While reading, the paper appears to be quite complex to follow, the artifact provides some helpful insights, and visual cues on the concrete usefulness of the work beyond the mathematical approach. In that sense, it is really helpful.

**As complete as possible**

The authors took care to provide the source code for all the 4 use cases and have carefully documented the system’s requirements. The current required configuration seems a bit limited (windows 10 pro edition). Unfortunately, I was not able to test everything, due to license issues, and with the exact configuration requirements mentioned by the authors. Therefore, the reported performance execution time during my tests are irrelevant, still the artifact is exercisable.

**Well-documented**

All aspects of the artifacts are documented. However, additional effort could have been put to describe the artifact into a consistant flow. For instance, rationale for the presence of specific materials, reference to the portion of the paper it refers to should be elaborated more carefully.

My main concern is related to the relationship with tale 10 of the paper, which is quite hard to understand between the different elements reported in the paper, which is quite well documented in the artifact documentation and when looking at the code in an IDE.
Some minor issues to manage (java heap space, etc.)

Comment: I have been running the spectral clusterer running example using the same picture (picture of a highway exit on google or from the authors) and having different results over several iterations. The authors should describe the expected outcomes of the running examples and code executions in order to provide guidance over the different executions and their interpretation.

** Easy to (re)use
Since there are licensing issues and platform requirements, the current artifact is currently difficult to fully reuse in a straightforward way.
We should praise the authors for efficiently separating the artifact material regarding their use, however, the artifact lacks a bit of description that would efficiently guide the reader through the webpage and through the material.

vonwenckstern commented 3 days ago

@AEReview I have been running the spectral clusterer running example using the same picture -> for this problem @sschneiders will contact you tomorrow via E-Mail. (@sschneiders has no rights to comment this issue here).

manuellerduc commented 2 days ago

Summary
The submitted artifact is part of a paper presenting an approach to optimize the compilation process of Component and Connector (C&C) models to many low-level compilation targets (e.g. x86/x64, arm, WebAssembly).
This approach is evaluated by comparing the execution time of 50 experiments.
To do so four case study are implemented and measurements are performed to evaluate:

- the effect of algebraic optimization on runtime speed;
- the impact of math backends on runtime performance;
- compare the performance of the generated code with other frameworks;
- compare the performance of WebAssembly generated code against handwritten MathJS code;
- compare the model sizes of EMA, Simulink, OpenModelica, Matlab, and Javascript (MathJS).

Documented
The documentation is well written and very complete.
I appreciated the extra references to complementary tutorials, in addition to the benchmark artifacts.

Consistent
The submitted artifacts are relevant and exhaustive in regard to the evaluation discussed in the paper.
The artifact contribute to the results presented in the case study section of the paper.

Complete
Artifacts are provided for every evaluated use cases.
Many complementary examples and references to tutorials are provided to the user.

Exercisable
The documentation allows the benchmarks to be executed easily by simply following the instructions.
The results produced by the execution of the benchmarks are comparable to the results presented in the paper.
However, the reusability is threatened by the support of Windows 10 only.

Availability

- Short-time use is certainly enable by the good documentation.
- Long-term use is threatened by the absence of a publication to a publicly accessible archival repository (I suggest http://remodd.org/ or https://zenodo.org/ but they are certainly not the only acceptable archival repositories).

Minor comments
The documentation can be improved by integrating a explicit inventory of the provided artifacts, as advised here https://www.acm.org/publications/policies/artifact-review-badging.

The reusability of the artifact might have been improved by pre-installing everything in a virtual machine.

vonwenckstern commented 2 days ago • edited

https://github.com/modelsconf2018/artifact-evaluation/issues/8
@manuelleduc I will upload all the HTML files and other files of our institute web storage for this publication to remodd.org (I will do it about 10:00 pm German time as I am right now in the train -- I hope this works still fine for you).

The reason why we did not do it yet is that the reader and you as reviewer then cannot browse through the models and inspect them online. - We put a lot of effort in our tooling to generate static HTML documentation models which are easy for inspection.

Internally we have a virtual machine with Windows 10 and our tooling, but I am afraid of that we get license issues when we just upload this Virtual Machine. That's why we packed everything to ZIP-files as portable apps so that you only need to unpack it and execute. We splitted the artifacts into several archives (for each benchmark class one) so that you only need to download the archive you are interested in. But if wanted, we can also pack all archives together into one large archive.

@AEReview
First, we would like to thank you for reporting that issue with the SpectralClusterer.

We were able to reproduce this issue with the online version of the SpectralClusterer.

In EMAStudio, which uses the native version of the SpectralClusterer, the results do not change if the same image is clustered several times. This issue indicates that there is a bug regarding the transformation of the C++ code to Webassembly.

So the prototype of the C++ to Webassembly toolchain part still needs some improvements.

I could also send you a VM image that contains EMAStudio, in case you prefer using a virtual machine.

EvgenyKusmenko commented 2 days ago

Dear all, we have uploaded our artifacts to https://zenodo.org/record/1314370#.W09hvNiizY2w

@grammarware @AEReview @manuelleduc

vonwenckstern commented 2 days ago

@EvgenyKusmenko @grammarware @manuelleduc I also added the archive link to the description of the readme.md file https://github.com/modelsconf2018/artifact-evaluation/blob/master/kusmenko/README.md

I hope we could so address your concerns about the long-term availability. -- Michael

md2manoppello commented 2 days ago • edited

The artifacts provided are part of the paper and present an approach to optimize the compilation process of Component and Connector (C&C) models to many low-level compilation targets.

The main contributions of this work are the following parts:

- a web-based version of the EmbeddedMontiArc solution,
- screencasts,
- materials and source code used for the case study, and
- a well-detailed list of actions to play with EmbeddedMontiArc compiler in a user-friendly environment

(1) Is the artifact consistent with the paper?
The artifacts are consistent with the paper since the supporting materials contribute to the results presented in the case study section of the paper.

(2) Is the artifact as complete as possible?
Artifacts are provided for all the 4 evaluated use cases. Moreover, the authors provide EmbeddedMontiArc framework that permits the execution of use case directly in a browser.

(3) Is the artifact well-documented?
The documentation is well written and very complete. Moreover, the source code of use cases has been carefully documented.

At the reference site (http://www.se-rwth.de/materials/ema_compiler/) many complementary examples and references to tutorials are provided to the user. The site also includes tutorials as screencasts.

(4) Is the artifact easy to (re)use?
By following the list of actions that the authors provide in the readme file, I can easily reuse the artifacts. Just one issue is related to the supported configuration: it is limited to Windows 10 pro edition.
Dear @vonwenckstern and @sschneiders,

Based on all the comments and the reviews provided by the members of the Artifact Evaluation Committee of MoDELS 2018, we have reached the conclusion that this artifact conforms to the expectations and is hereby approved. Please use the badge instructions page to add the badge of approval to your article, and add the link to the Zenodo entry with DOI https://doi.org/10.5281/zenodo.1314370 to the camera ready version of the paper.

Thank you very much for putting extra effort into the preparation and finalising of the artifact. If any of the comments above are still not addressed, please try to accommodate them before the conference.