Parallelizing Julia with a Non-Invasive DSL (Artifact)

Todd A. Anderson[1], Hai Liu[2], Lindsey Kuper[3], Ehsan Totoni[4], Jan Vitek[5], and Tatiana Shpeisman[6]

1 **Parallel Computing Lab, Intel Labs**
2 **Parallel Computing Lab, Intel Labs**
3 **Parallel Computing Lab, Intel Labs**
4 **Parallel Computing Lab, Intel Labs**
5 **Northeastern University / Czech Technical University Prague**
6 **Parallel Computing Lab, Intel Labs**

Abstract

This artifact is based on ParallelAccelerator, an embedded domain-specific language (DSL) and compiler for speeding up compute-intensive Julia programs. In particular, Julia code that makes heavy use of aggregate array operations is a good candidate for speeding up with ParallelAccelerator. ParallelAccelerator is a *non-invasive* DSL that makes as few changes to the host programming model as possible.

1998 ACM Subject Classification D.1.3 Parallel Programming

Keywords and phrases parallelism, scientific computing, domain-specific languages, Julia

Digital Object Identifier 10.4230/DARTS.3.2.7

Related Article Todd A. Anderson, Hai Liu, Lindsey Kuper, Ehsan Totoni, Jan Vitek and Tatiana Shpeisman, "Parallelizing Julia with a Non-Invasive DSL", in Proceedings of the 31st European Conference on Object-Oriented Programming (ECOOP 2017), LIPIcs, Vol. 74, pp. 4:1–4:29, 2017. `http://dx.doi.org/10.4230/LIPIcs.ECOOP.2017.4`

Related Conference European Conference on Object-Oriented Programming (ECOOP 2017), June 18-23, 2017, Barcelona, Spain

1 Scope

This artifact allows the user to install, run, test, and benchmark the ParallelAccelerator Julia package. It is designed to support repeatability of the experiments of the companion paper.

2 Content

This artifact contains:

- Snapshots of the most recent tagged releases of the ParallelAccelerator (v0.2.2) and CompilerTools (v0.2.1) Julia packages at the time of artifact release. In addition to source code for the ParallelAccelerator compiler itself, the ParallelAccelerator package also includes the code for implementations (using both ParallelAccelerator and plain Julia) of all the workloads discussed in the companion paper, as well as for a few additional workloads.
- Additional MATLAB, Python, and C/C++ implementations of the workloads discussed in the companion paper, as well as code for benchmarking and plotting of benchmarking results.
- Detailed instructions for using the artifact, provided as a `README.pdf` file.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The latest version of ParallelAccelerator is available on GitHub at `https://github.com/IntelLabs/ParallelAccelerator.jl`.

4 Tested platforms

ParallelAccelerator requires a *nix OS, ideally Linux. Platforms we have tested on include Ubuntu 16.04, Ubuntu 14.04, CentOS 6.6, macOS Yosemite, and macOS Sierra.

5 License

See `https://github.com/IntelLabs/ParallelAccelerator.jl/blob/master/LICENSE.md` for the ParallelAccelerator license.

6 MD5 sum of the artifact

c8eba9e27a8c6c2b45612c883e20bbac

7 Size of the artifact

54.97 MB

Acknowledgements. We thank Gabriel Scherer for testing, and the ECOOP '17 Artifact Evaluation Committee reviewers for their helpful comments.