Optimization Coaching for JavaScript (Artifact)

Vincent St-Amour[1] and Shu-yu Guo[2]

1 **PLT @ Northeastern University**
Boston, Massachusetts, USA
stamourv@ccs.neu.edu
2 **Mozilla Research**
San Francisco, California, USA
shu@mozilla.com

Abstract

This artifact is based on our prototype optimization coach for the SpiderMonkey[1] JavaScript engine. An optimization coach is a performance tool that aims to provide programmers with insight into how their compiler optimizes their programs and to help them better harness the optimization process. It does so by reporting *optimization near misses*, i.e., reports of optimizations that the compiler did not apply, but could apply if the program were to be modified slightly.

This artifact provides the necessary environment, programs and data to repeat our experiments, and to allow readers to run our tool on JavaScript programs of their choice.

1998 ACM Subject Classification D.2.3 [Software Engineering]: Coding Tools and Techniques, D.3.4 [Programming Languages]: Processors – Compilers

Keywords and phrases Optimization Coaching, JavaScript, Performance Tools

Digital Object Identifier 10.4230/DARTS.1.1.5

Related Article Vincent St-Amour and Shu-yu Guo, "Optimization Coaching for JavaScript", in Proceedings of the 29th European Conference on Object-Oriented Programming (ECOOP 2015), LIPIcs, Vol. 37, pp. 271–295, 2015.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2015.271

Related Conference 29th European Conference on Object-Oriented Programming (ECOOP 2015), July 5–10, 2015, Prague, Czech Republic

1 Scope

The goal of this artifact is to support repeating the experiments presented in our paper. These experiments include our performance experiments, which compare baseline and post-coaching versions of well-known JavaScript benchmarks, and the coaching experiments themselves, which involve generating and following the tool's recommendations. In addition, this artifact allows interested readers to run our optimization coach on programs of their choice.

2 Content

The artifact package includes:

- Our prototype coach.
- A version of the SpiderMonkey JavaScript engine, instrumented to gather optimization information (based on revision `f0f846d875acaced`, September 2014).
- Unmodified versions of the JavaScriptCore (Version 2.4.6, September 2014) and V8 (Version 3.30.5, October 2014) JavaScript engines.

[1] https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey

- The benchmark programs we used, with both pre- and post- coaching versions.
- Pre-generated profiling and coaching output, as well as the means to generate your own.

The artifact is packaged as a Debian GNU/Linux virtual machine. To log in to the virtual machine, use the following credentials:

- username: **artifact**
- password: **artifact**
- root password: **artifact**

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 source code of our prototype is available on GitHub:
`https://github.com/stamourv/jit-coach/`

The instrumented version of SpiderMonkey it uses is also available on GitHub:
`https://github.com/stamourv/gecko-dev/tree/profiler-opt-info`

4 Tested platforms

This artifact has been tested on both GNU/Linux and Mac OS X, using Oracle VirtualBox version 4. About 5 GB of free space on disk is required, and at least 4 GB of RAM is recommended.

5 License

MPL 2.0 (`https://www.mozilla.org/MPL/`)

6 MD5 sum of the artifact

234f2635bd752a19c8d231fe95ea8574

7 Size of the artifact

4.3 GB

Acknowledgements. We would like to thank Niko Matsakis, Dave Herman, and Michael Bebenita for discussions and suggestions about the tool's design and development. Kannan Vijayan, Luke Wagner, and Nicolas Pierron helped with the design of the profiler-driven instrumentation.