

Publications and Presentations

Graal

The Graal publications and presentations are now at <https://github.com/graalvm/graal-core/blob/master/docs/Publications.md>

Truffle Tutorial: One VM to Rule Them All, One VM to Bind Them

Forget “this language is fast”, “this language has the libraries I need”, and “this language has the tool support I need”. The Truffle framework for implementing managed languages in Java gives you native performance, multi-language integration with all other Truffle languages, and tool support - all of that by just implementing an abstract syntax tree (AST) interpreter in Java.

Truffle applies AST specialization during interpretation, which enables partial evaluation to create highly optimized native code without the need to write a compiler specifically for a language. The Java VM contributes high-performance garbage collection, threads, and parallelism support.

This tutorial is both for newcomers who want to learn the basic principles of Truffle, and for people with Truffle experience who want to learn about recently added features. It presents the basic principles of the partial evaluation used by Truffle and the Truffle DSL used for type specializations, as well as features that were added recently such as the language-agnostic object model, language integration, and debugging support.

Oracle Labs and external research groups have implemented a variety of programming languages on top of Truffle, including JavaScript, Ruby, R, Python, and Smalltalk. Several of them already exceed the best implementation of that language that existed before.

PLDI 2016, June 13, 2016, Santa Barbara, CA

[Video recording](#)

[Download slides](#)

Truffle Presentations

- JVM Language Summit 2014, July 28-30, Santa Clara, CA: *Debugging at Full Speed: Instrumenting Truffle-implemented Programs*

[Video recording](#)

[Download slides](#)

- JVM Language Summit 2013, July 29-31, Santa Clara, CA: *One VM to Rule Them All*

[Video recording](#)

[Download slides](#)

- SPLASH 2012, October 19-26, Tucson AZ: *Truffle: A Self-Optimizing Runtime System*. [doi:10.1145/2384716.2384723](https://doi.org/10.1145/2384716.2384723)

[Download slides](#)

[Download abstract](#)

Truffle Papers

- Matthias Grimmer, Chris Seaton, Thomas Würthinger, Hanspeter Mössenböck: *Dynamically Composing Languages in a Modular Way: Supporting C Extensions for Dynamic Languages*. accepted for publication at the *14th International Conference on Modularity*, 2015
- Gülfem Savrun-Yeniçeri, Michael Van De Vanter, Per Larsen, Stefan Brunthaler, and Michael Franz, *An Efficient and Generic Event-based Profiler Framework for Dynamic Languages*
In the *2015 International Conference on Principles and Practices of Programming on the Java Platform: virtual machines, languages, and tools* (PPJ2015), September 9, 2015.
- Michael Van De Vanter, *Building Debuggers and Other Tools: We Can "Have it All"* (Position Paper)
In the *10th Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems Workshop* (ICOOOLPS 2015), July 6, 2015.

[Download paper](#)

- Matthias Grimmer: *High-performance language interoperability in multi-language runtimes*.
In *Proceedings of the companion publication of the 2014 ACM SIGPLAN conference on Systems, Programming, and Applications: Software for Humanity (SPLASH'14)*, 2014

[Download paper](#)

- Matthias Grimmer, Manuel Rigger, Roland Schatz, Lukas Stadler, Hanspeter Mössenböck: *Truffle C: Dynamic Execution of C on the Java Virtual Machine*.
In *Proceedings of the Intl. Conf. on Principles and Practice of Programming in Java (PPPJ'14)*, 2014

[Download paper](#)

- Christian Humer, Christian Wimmer, Christian Wirth, Andreas Wöss, Thomas Würthinger: *A Domain-Specific Language for Building Self-Optimizing AST Interpreters*. In *Proceedings of the International Conference on Generative Programming: Concepts and Experiences (GPCE'14)*, 2014

[Download paper](#)

- Andreas Wöss, Christian Wirth, Daniele Bonetta, Chris Seaton, Christian Humer, Hanspeter Mössenböck: *An Object Storage Model for the Truffle Language Implementation Framework*. In *Proceedings of Intl. Conf. on Principles and Practice of Programming in Java (PPPJ'14)*, 2014

[Download paper](#)

- Matthias Grimmer, Thomas Würthinger, Andreas Wöss, Hanspeter Mössenböck: *An Efficient Approach to Access Native Binary Data from JavaScript*. In *Proceedings of the 9th Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS2014)*, 2014

[Download paper](#)

- Thomas Würthinger, Christian Wimmer, Andreas Wöss, Lukas Stadler, Gilles Duboscq, Christian Humer, Gregor Richards, Doug Simon, Mario Wolczko: *One VM to Rule Them All*. In *Proceedings of Onward!*, ACM Press, 2013. doi:10.1145/2509578.2509581

[Download paper](#)

Describes the vision of the Truffle approach, and the full system stack including the interpreter and dynamic compiler.

- Chris Seaton, Michael Van De Vanter, and Michael Haupt: *Debugging at full speed*. In *Proceedings of the 8th Workshop on Dynamic Languages and Applications (DYLA)*, 2014.

[Download paper](#)

- Matthias Grimmer, Manuel Rigger, Lukas Stadler, Roland Schatz, Hanspeter Mössenböck: *An efficient native function interface for Java*. In *Proceedings of the International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools. (PPPJ'13)*, 2013

[Download paper](#)

- Matthias Grimmer: *Runtime Environment for the Truffle/C VM*. Master's thesis, Johannes Kepler University Linz, November 2013

[Download](#)

- Thomas Würthinger, Andreas Wöss, Lukas Stadler, Gilles Duboscq, Doug Simon, Christian Wimmer: *Self-Optimizing AST Interpreters*. In *Proceedings of the Dynamic Languages Symposium*, pages 73-82. ACM Press, 2012. doi:10.1145/2384577.2384587

[Download paper](#)

Describes the design of self-optimizing and self-specializing interpreter, and the application to JavaScript.