

Zero-overhead R and C/C++ integration with FastR

Lukas Stadler, Štěpán Šindelář, Florian Angerer
Oracle Labs

The Problem

C and C++ are frequently used to improve performance of R applications and packages. While this is usually not necessary when using FastR, because it can run R code at near-native performance, there is a large corpus of existing code that implements critical pieces of functionality in native code. Like other alternative implementations of R, FastR needs to simulate the R native API, which is a complex API that exposes many implementation details. This simulation usually incurs significant effort and performance overhead, and there is a compilation and optimization barrier between languages.

The Solution

FastR uses the Truffle framework and the Sulong engine to run native code, available as LLVM bitcode, inside the optimization scope of the polyglot Truffle environment. This allows the Graal compiler to perform its advanced compiler optimizations for both languages and across both languages.

FastR

FastR is an alternative implementation of the R language, running on top of a Java Virtual Machine. It is designed to be a drop-in replacement that executes existing R code at unparalleled peak performance. It also provides access to the polyglot Truffle ecosystem, so that it can interact efficiently with other languages such as JavaScript and Ruby.

Sulong

Sulong is an interpreter for LLVM IR code that can execute C/C++, Fortran, and other LLVM-supported languages on a Java Virtual Machine. FastR can use Sulong to execute code written in these languages within the same ecosystem, compile them using the same compiler, and optimize them as a single unit for maximum performance.

Truffle

Truffle is a framework for implementing languages as simple interpreters. It provides the basic foundation for building abstract-syntax-tree (AST) interpreters that perform self-optimization at runtime. The included TruffleDSL provides a convenient way to express such optimizations.

Truffle is developed and maintained by Oracle Labs and the Institute for System Software of the Johannes Kepler University Linz.

Graal

Graal is a dynamic compiler that transforms Java bytecode into executable machine code. It is written in Java, and integrates into Java Virtual Machines like the HotSpot JVM. It has a focus on high peak performance, maintainability and extensibility.

Together with the Graal compiler, Truffle is capable of just-in-time compiling programs running on top of it to native code for efficient execution.

