JIT Compilation in V8 JS Engine

Jan Kubiš (xkubis13@stud.fit.vutbr.cz)
Jaroslav Holcman (xholcm04@stud.fit.vutbr.cz)

October 24, 2017

Abstract

V8 is a JavaScript engine developed by Google. The implementation is written in C++ and is completely open-source. It is designed to improve the performance of JavaScript code in web browsers. As opposed to classic approach of JavaScript execution, V8 compiles the JavaScript code directly into native machine code without producing any intermediate code. Another feature is that the compilation is dynamic - it is done during runtime (just-in-time). This approach turned out to be much faster and efficient in comparison with common techniques, which usually produce an intermediate code for internal representation.

V8 consists of two compilers - full compiler and optimizing compiler. The full compiler compiles all used parts of code into machine code and stores it in the inline cache (IC). Its goal is to compile as quickly as possible, so it does not care about any optimizations. The optimizing compiler then takes the most used functions and optimizes just them.

In our presentation, we are going to describe most significant mechanisms which V8 uses. One of them is the usage of hidden classes. Hidden classes are used in V8 to enable sharing of the same optimized machine code between more objects. Finally, we will talk about ideas such as inline-cache, fast properties, deoptimization, constant function descriptors and in-object slack tracking.