Comparison of Compilation Methods

Jim Mouer
Department of Computer Science, Furman University, Greenville, SC

What is a compiler?

- A compiler is a piece of software that converts a program written in a high-level programming language into instructions that the processor can execute.
- Compilers perform many optimizations on code while translating it, such as unrolling loops or converting complex mathematical operations into groups of simpler operations.
- Most compilers can be divided into the category of ahead-of-time or just-in-time. A related third category is the interpreter, which is not a compiler but has similarities to a just-in-time compiler.

Ahead-of-Time Compilation

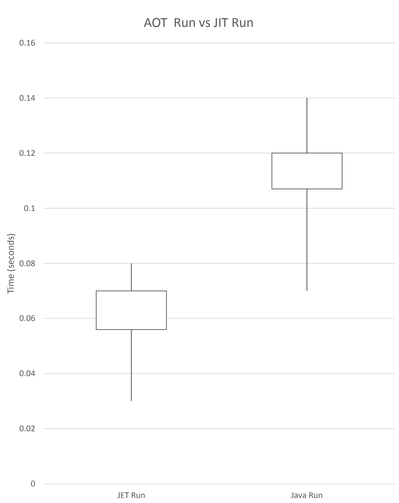
- Ahead-of-time compilation (or AOT) refers to the practice of running a separate compilation command prior to runtime, generating a separate file that is then executed by the computer.
- Programs that are compiled AOT generate machine code that is specific to one computer or type of computer. In order to be run on a different computer, the original code must be recompiled.
- Compiled code may be run on that machine any number of times without recompiling.
- Programs that are compiled AOT tend to run faster than their just-in-time counterparts, since there is more time available to optimize code.
- Notable languages that use AOT compilation include C, C++, and Pascal.

Just-in-Time Compilation

- Just-in-Time compilation (or JIT) is performed when a user or another program calls the program to run.
- Since compilation is performed at runtime, programs using JIT often have higher startup overhead than those using AOT.
- JIT compilers have access to some data about the program at runtime, such as the values of variables, that can help make optimizations that an AOT compiler cannot, such as more advanced branch prediction.
- Notable languages that use JIT compilation include Java and C#.

Comparison

- In order to compare the two compilation methods, I compiled and ran a short program using both methods.
- For my AOT compiler, I used Excelsior JET, a proprietary AOT Java compiler.
- For my JIT compiler, I used the Java Standard Edition Runtime Environment (JRE).
- To time the compilation of Excelsior JET, I used the program's internal compilation timer. To time the compilation of the JRE and the execution of both, I used the Linux "time" command.
- In order to account for threading on my multi-core processor, I measured the timings several times and took the mean values.



Challenges

- Technically, the Java language uses both AOT and JIT. It uses AOT to convert the Java code into Java bytecode, an intermediary language, which is then run with the JIT compiler. This makes it more difficult to fairly compare the two. To account for this, "Java Compile and Run" includes this time on the graph.
- It was difficult to find a way to compare AOT and JIT in the same language. There are not many languages that have both types of compilers built, much less maintained.
- My results represent neither a typical nor an extraordinary usage of either method. Both may perform better or worse in other implementations.

Additional Work

There is significant room for additional work in this area. This work includes:

- Comparisons using other programming languages.
- More thorough and varied tests, rather than a single program.
- Robust automated testing.
- Analysis of structures on which each method excels.

References

John Aycock. 2003. A brief history of just-in-time. ACM Comput. Surv. 35, 2 (June 2003), 97-113. DOI=http://dx.doi.org/10.1145/857076.857077

Bucciarelli, Thiemo. "Just-In-Time Compilation." Universität Zu Lübeck. Web.

V. Mikheev, N. Lipsky, D. Gurchenkov, P. Pavlov, V. Sukharev, A. Markov, S. Kuksenko, S. Fedoseev, D. Leskov, and A. Yeryomin. 2002. Overview of excelsior JET, a high performance alternative to java virtual machines. In Proceedings of the 3rd international workshop on Software and performance (WOSP '02). ACM, New York, NY, USA, 104-113. DOI=http://dx.doi.org/10.1145/584369.584387