Contribution ID: **158** Type: **not specified**

Lightning Talk: Teaching GraalVM DWARFish : Debugging Native Java using gdb

Monday 24 August 2020 08:15 (10 minutes)

Or is it DWARVish? Whatever, GraalVM Native implements compilation of a complete suite of Java application classes to a single, complete, native ELF image. It's much like how a C/C++ program gets compiled. Well, except that the image contains nothing to explain how the bits were derived from source types and methods or where those elements were defined. Oh and the generated code is heavily inlined and optimized (think gcc -O2/3). Plus many JDK runtime classes and methods get substituted with lightweight replacements. So, a debugging nightmare.

Anyway, we have resolved the debug problem much like how you do with C/C++ by generating DWARF records to accompany and explain the program bits. So far, we have file and line number resolution, breakpoints, single stepping & stack backtraces. We're now working on type names and layouts and type, location & liveness info for heap-mapped values/objects, parameters and local vars. I'll explain how we obtain the necessary input from the Java compiler, how we model it as DWARF records and how we test it for correctness using objdump and gdb itself. By that point I will probably need to stop to take a breath

I agree to abide by the anti-harassment policy

I agree

Primary author: DINN, Andrew (Red hat)

Presenter: DINN, Andrew (Red hat)

Session Classification: GNU Tools Track

Track Classification: GNU Tools Track