

Method Invocation in Templated JIT Compiler

Harpreet Kaur, Kenneth B. Kent

Faculty of Computer Science, University of New Brunswick

Marius Pirvu

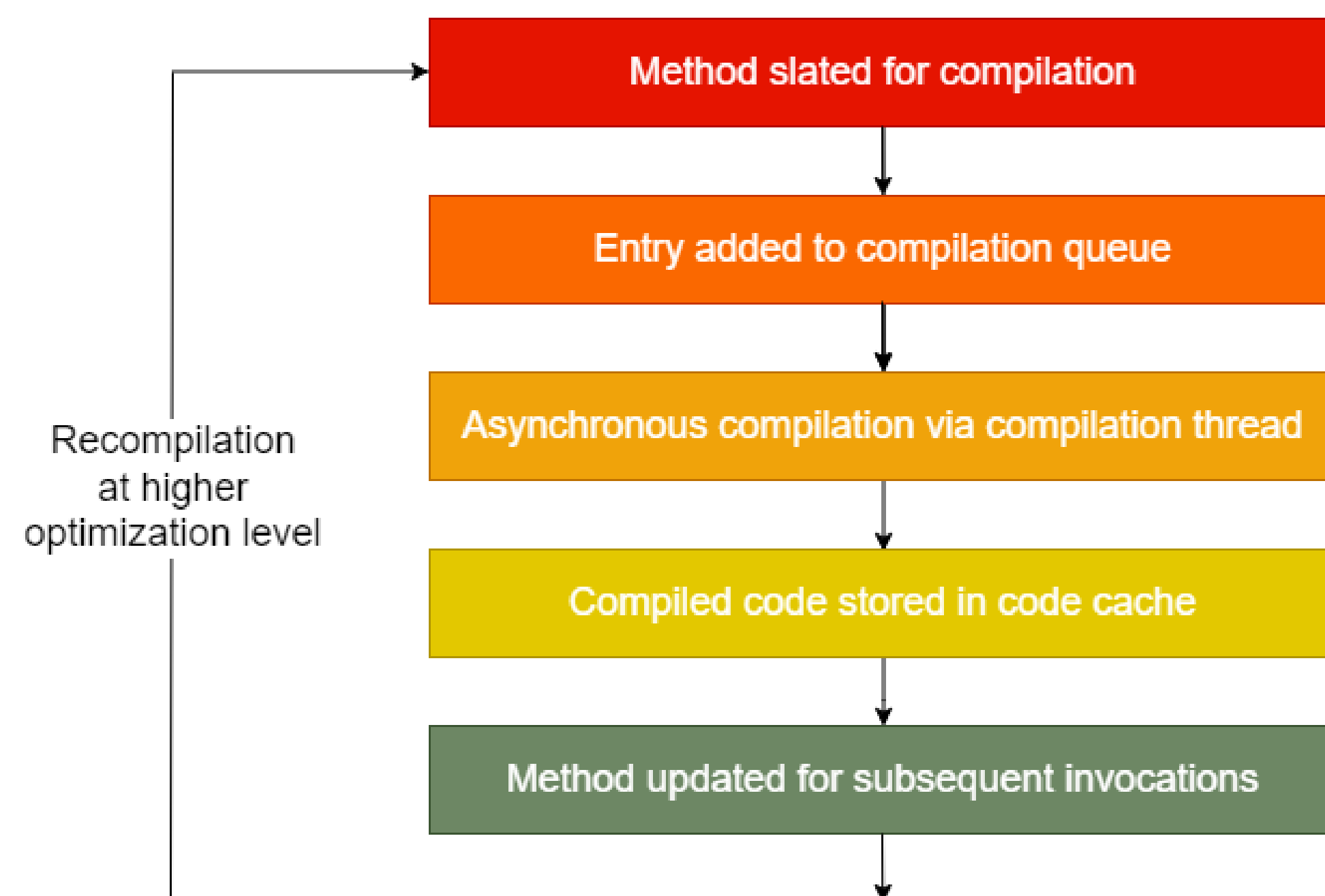
IBM Canada

{harpreet.bamrah, ken}@unb.ca

mpirvu@ca.ibm.com

Background

- IBM CAS Project 1038 – MicroJIT for OMR
- Eclipse OpenJ9 is an open-source, high performance Java Virtual Machine (JVM) implementation:
 - Built on top of core technologies provided by Eclipse OMR
- Just-in-Time (JIT) compilation is a technique for optimizing language runtimes
- Optimizing compilers maximize throughput at the cost of memory footprint and compilation time
- Templated compilers generate native code more quickly
- Compilation lifecycle:



MicroJIT

- Lightweight, non-optimizing JIT compiler
- Expands upon JVM implementation of Eclipse OpenJ9 for Java 8 on x86-64 Linux platform
- Promising results for Eclipse OpenJ9 in resource constrained environments
- NASM assembly template for *iadd* bytecode:

```
template_start iAddTemplate
xor r11, r11 ; clear r11
xor r12, r12 ; clear r12
mov r11, [r10] ; pop first value off java stack into the accumulator
add r10, 8 ; which means reducing the stack size by 1 slot (8 bytes)
mov r12, [r10] ; copy second value to the value register
add r11d, r12d ; add the value to the accumulator
mov [r10], r11 ; write the accumulator over the second arg.
template_end iAddTemplate
```

Method Invocation

- Powerful tool for JVM languages
- Java method calls converted to one of several method invocation bytecodes
- MicroJIT signals compilation failure for unsupported bytecodes
- Implementing these will improve the JVM performance

Bytecode	Purpose	Implemented
<i>invokestatic</i>	Call static methods	Yes
<i>invokevirtual</i>	Invoke methods via virtual dispatch (polymorphic calls)	Yes
<i>invokespecial</i>	Handle special cases	Yes
<i>invokeinterface</i>	Invoke interface methods	No
<i>invokedynamic</i>	Dynamic invocations	No

Next Steps

- Materialize the prototype MicroJIT implementation:
 - Pull Request in OpenJ9: github.com/eclipse-openj9/openj9/pull/9578
 - Pull Request in OpenJDK8: github.com/ibmruntimes/openj9-openjdk-jdk8/pull/413
- Low-level optimizations, like replacing virtual calls with direct method calls in case of monomorphic call sites
- Enhancing bytecode support:
 - 78% as on date
- Exception handling in MicroJIT
 - Using *athrow* instruction
- Compiling switches
- Extending MicroJIT to other architectures:
 - Currently supports x86-64

