

OpenJDK for RV32G的 解释器与C2

中科院软件所 史宁宁
shiningning@iscas.ac.cn
2022年12月9日

目录

- 进度与现状
- C2的基本情况
- C2的移植和调试
- 其他

目录

- 进度与现状
- C2的基本情况
- C2的移植和调试
- 其他

进度与现状

- 2022年1月11日，OpenJDK 11 for RV32G的模板解释器跑通了“Hello world”。
- 2022年2月25日，OpenJDK 11 for RV32G的模板解释器已经完成，开始移植OpenJDK 11 for RV32G的C2编译器。
- OpenJDK 11 for RV32G的当前情况：
 - OpenJDK 11 for RV32G的模板解释器可以支持绝大多数测试集中的测试用例，SPECjvm2008 100%，jctstress 99%，DaCapo 78%，jreg 76%，具体可以查看<https://github.com/openjdk-riscv/jdk11u/issues/335>；
 - OpenJDK 11 for RV32G的C2 仍然在debug中，可以编译通过，但是运行时还有bug；
 - OpenJDK 11 for RV32G的C1 处于TODO状态。
- 团队状况：2022年1月1日——2022年7月31日，史宁宁、张定立、章翔、曹贵；2022年8月1日至今，史宁宁。

项目基本信息

- PLCT实验室目前所进行的RV32G工作，工作过程和工作产出都在github公开：<https://github.com/openjdk-riscv/jdk11u>
- 在移植过程中，PLCT实验室产出了几十篇技术文章和技术报告，这些内容分别公开在：
 - 1) <https://github.com/openjdk-riscv/jdk11u/wiki>
 - 2) https://www.zhihu.com/column/c_1287750038518161408
 - 3) <https://space.bilibili.com/296494084/video?keyword=openjdk>
- 个人专栏：
OpenJDK 点点滴滴：
https://www.zhihu.com/column/c_1567556501417541632

目录

- 进度与现状
- C2的基本情况
- C2的移植和调试
- 其他

解释器与C2

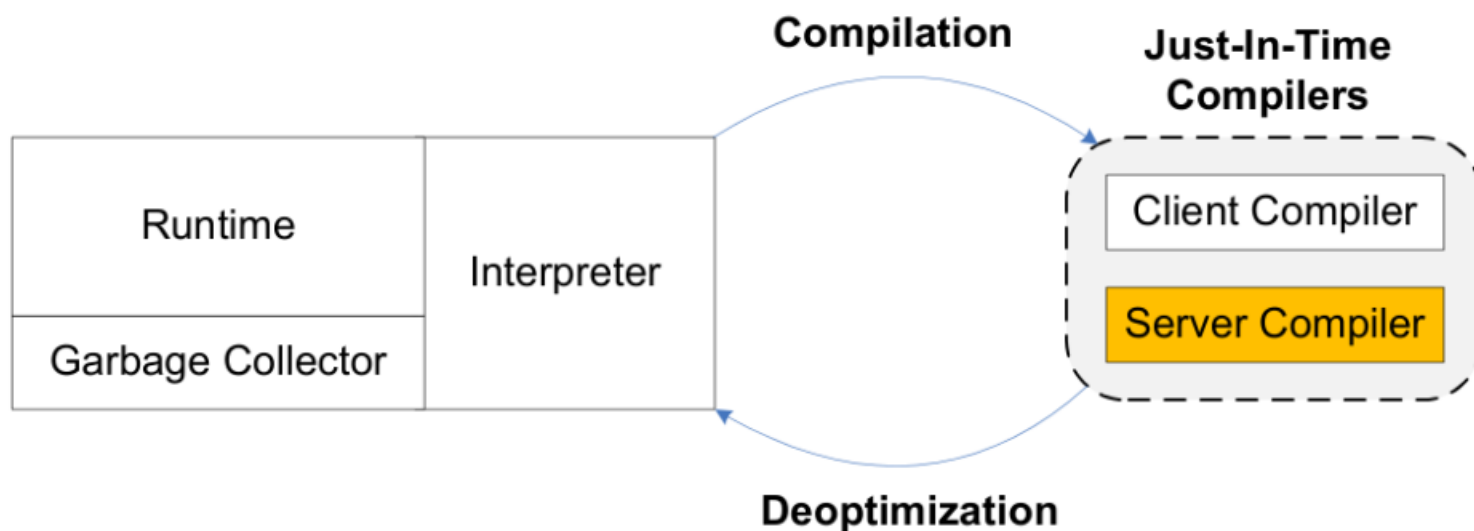
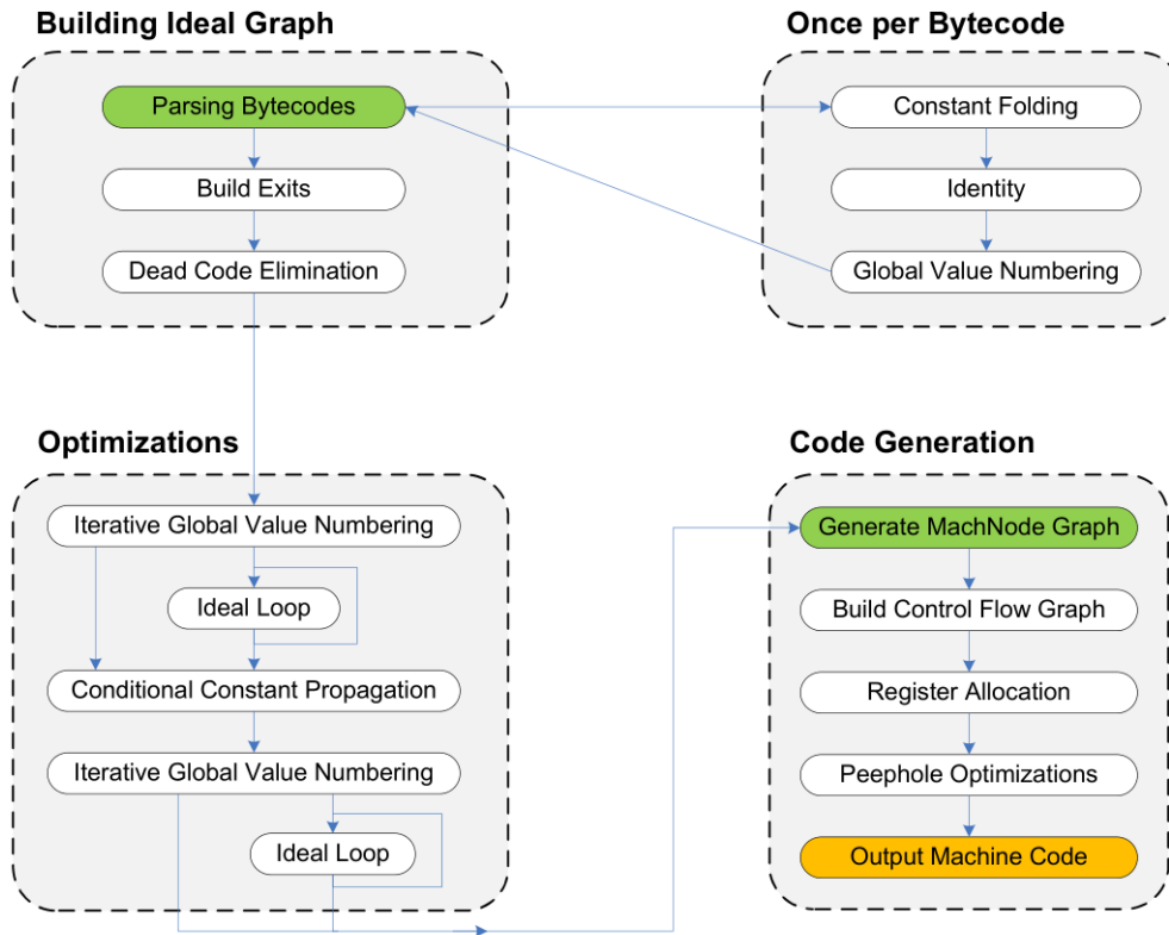


Figure 3.1: Architecture of the Java HotSpot™ Virtual Machine.

From: 《Visualization of Program Dependence Graphs》 P13

C2的基本流程



From: 《
Visualization
of Program
Dependence
Graphs
》 P16

Figure 3.3: Architecture of the Java HotSpot™ server compiler of Sun Microsystems.

C2 IR 例子1

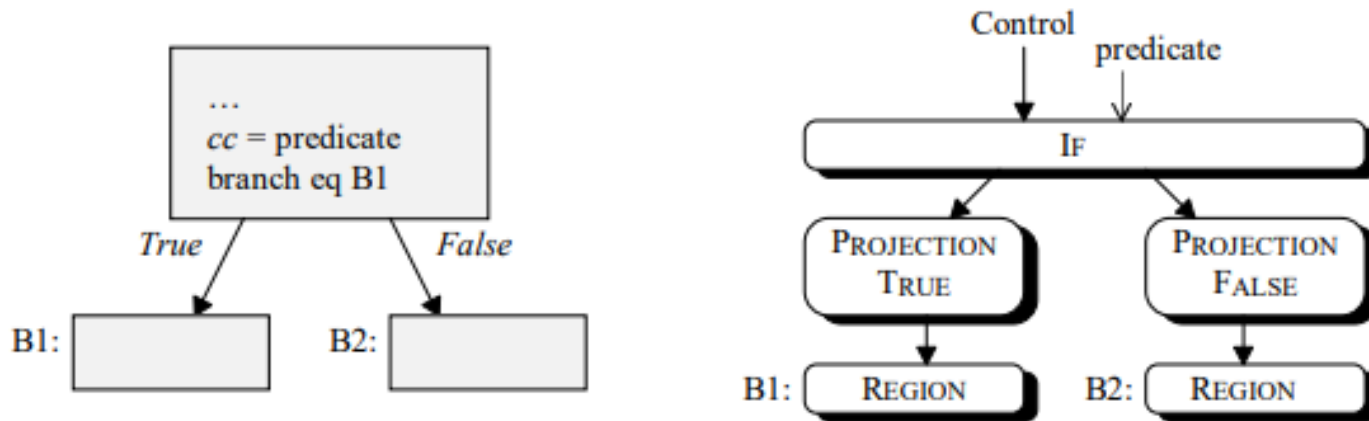


Figure 5 Projections following an IF Node

From: 《A Simple Graph-Based Intermediate Representation》 P38

C2 IR 例子2

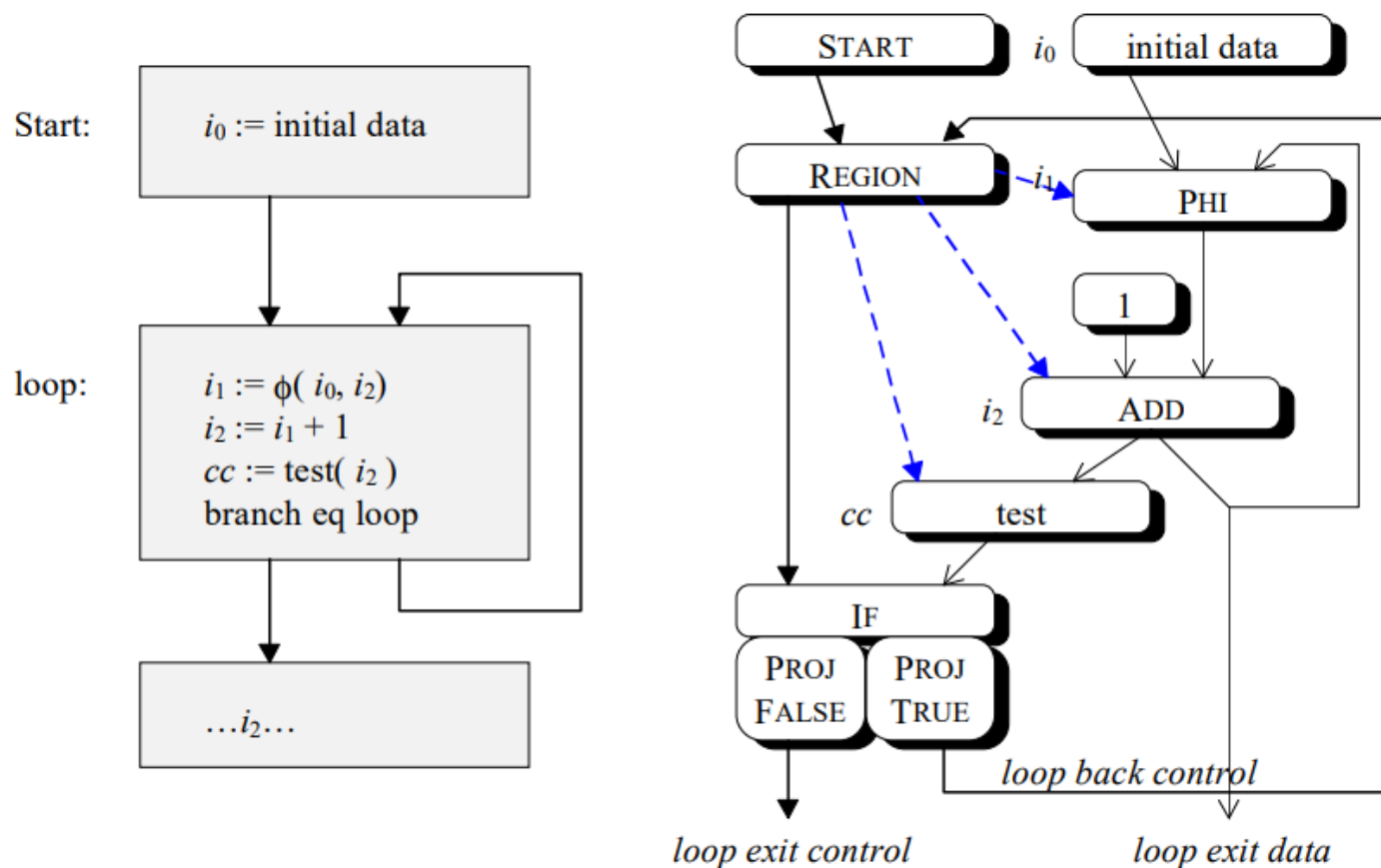


Figure 7 An example loop

From: 《A Simple Graph-Based Intermediate Representation》 P40

目录

- 进度与现状
- C2的基本情况
- C2的移植和调试
- 其他

C2的移植——AD文件

- AD文件通常会使用ADL描述一个架构的三个基本信息：指令集、寄存器组和流水线信息，这三个信息被用在编译器后端的三个主要环节：指令选择、寄存器分配和指令调度中。AD文件会被交给ADL Compiler进行编译，生成对应的代码。
- RV32G涉及到的AD文件为：
`src/hotspot/cpu/riscv32/riscv32.ad`。

C2的移植——Long数据的处理

- Long的算术运算
- Long的逻辑运算
- Long的比较
- Long和其他数据类型的转换
- Long数据的存取和加载
-

C2的调试—调试参数

- -XX:+参数
- +LogCompilation
produces a low-level XML file about compiler and runtime decisions
- +PrintAssembly
print assembly code for bytecoded and native methods
- +PrintNMethods
print nmethods as they are generated
- +PrintNativeNMethods
print native method wrappers as they are generated
- +PrintSignatureHandlers
print native method signature handlers

From: <https://wiki.openjdk.org/display/HotSpot/PrintAssembly>

C2的调试—调试参数（续）

- +PrintAdapterHandlers
print adapters (i2c, c2i) as they are generated
- +PrintStubCode
print stubs: deopt, uncommon trap, exception, safepoint, runtime support
- +PrintInterpreter
print interpreter code

From: <https://wiki.openjdk.org/display/HotSpot/PrintAssembly>

C2的调试——调试工具

- **IdealGraphVisualizer**

The Ideal Graph Visualizer is a tool developed to help examine the intermediate representation of C2 which is commonly referred to as the "ideal graph". It was developed in collaboration with the University of Linz in Austria and is included in HotSpot's repository since that was the primary target of the tool. The tool itself is fairly general with only a few modules that contain C2 specific elements.

From: <https://wiki.openjdk.org/display/HotSpot/IdealGraphVisualizer>

目录

- 进度与现状
- C2的基本情况
- C2的移植和调试
- 其他

其他工作——出版物





计算机技术开发与应用丛书

- 深度解析Android Runtime架构与流程的实现
- 中国科学院软件研究所高级工程师邢明杰、OSDT社区负责人吴伟等联袂推荐

Android Runtime 源码解析

史宁宁◎编著

清华大学出版社

定价：79.00元 ISBN: 978-7-302-60084-8

方舟编译器社区

- 方舟编译器周报目前已经更新到第141期：
https://www.zhihu.com/column/c_1268247974020747264
- 方舟编译器开源三周年时，发布了总结性的文章：
 - 方舟编译器开源这三年，我写了哪些技术文章？
<https://zhuanlan.zhihu.com/p/551153441>
 - 方舟编译器开源这三年，我作了哪些技术分享？
<https://zhuanlan.zhihu.com/p/553289930>

致谢

- 华为Bisheng JDK团队 。
- 中科院软件所PLCT实验室：张定立、章翔、曹贵。
- 中科院软件所PLCT实验室实习生：陈家友。

参考文献

- Thomas Würthinger. Visualization of Program Dependence Graphs. Master Degree Paper. 2007.
<https://ssw.jku.at/Research/Papers/Wuerthinger07Master/Wuerthinger07Master.pdf>
- Cliff Click, Michael Paleczny. A Simple Graph-Based Intermediate Representation. 1995.
<https://www.oracle.com/technetwork/java/javase/tech/c2-ir95-150110.pdf>
- <https://wiki.openjdk.org/display/HotSpot/PrintAssembly>
- <https://wiki.openjdk.org/display/HotSpot/IdealGraphVisualizer>

Thanks!