计算机体系结构

Lab 分支预测的算法与实践

Assigned Mar.26,2018 Worth 5% Cost:3 Week

http://cs.hit.edu.cn

Summary

分支结构在流水线处理器中可能会创建指令冒险。当一个指令被加载时，处理器必须决定下一个要执行的指令。大多数情况下，x86 架构中的后续指令位于当前指令地址之后的字节中。因此只要取出下一个字节的内容并执行，就一个合理的策略。而对于分支指令，下一条指令可能位于内存的其他地方。在这种情况下直接加载下一条指令可能是错误的，并且当处理器回头读取正确的下一条指令时，流水线将会被停止（Stall）。

由于分支指令占指令总数的 1/6，因此错误地读取分支指令之后的下一条指令可能会对处理器性能造成极大的影响。设计师们用来处理指令冒险的机制是分支预测。在较高层次上，分支预测的接口很简单：分支预测器接受来自处理器的指令地址并返回其分支方向。后续指令地址的值由另一个结构（如 BTB）预测。请注意，一个很平庸的分支预测器实现，就是总返回一个“不跳转”的结果，即按顺序直接进入下一个指令。在本实验中，你将研究预测器并实现一个分支预测器模型。

在这个实验中，我们将使用 Pin 和 SPEC 基准测试程序来评估你的分支预测器的行为。

同之前的实验一样，这个实验需要你单独完成，你可以通过阅读附带的 Pin 指导内容来学习使用，或与同学讨论你遇到的问题。但是请注意按时提交你的成果。

Setting Up

实验包中附有实验模板 BranchPredictor.h，请在 Linux 环境下打开进行编辑（在入门中我们提及到，在 Windows 环境下进行编程及保存可能会改变文件信息格式，造成无法使用）。另外，lab.cpp 为本次实验的 Pintool 文件，你不需要修改它，而是可以通过利用这个 Pintool 调用并测试你的预测器。最终的结果输出格式已经定义好，同样不需要修改。

请将本次实验的两个资源文件（BranchPredictor.h、lab.cpp）和被测试文件（test.cpp）放置在你的 Pin 路径下的/source/tools/ManualExamples 中，这样生成的.so 文件就在 obj-intel64 中以便使用。请不要忘记在 makefiles.rules 中添加 lab 工具的引用，才能正常编译。其余设置均和入门保持一致。

编译好后，要想执行这个工具来测试你的预测器，命令：pin –t obj-intel64/lab.so –test.cpp，让你的预测器对 test.cpp 进行插桩测试，你也可以使用其他的被测试程序。

Lab Task

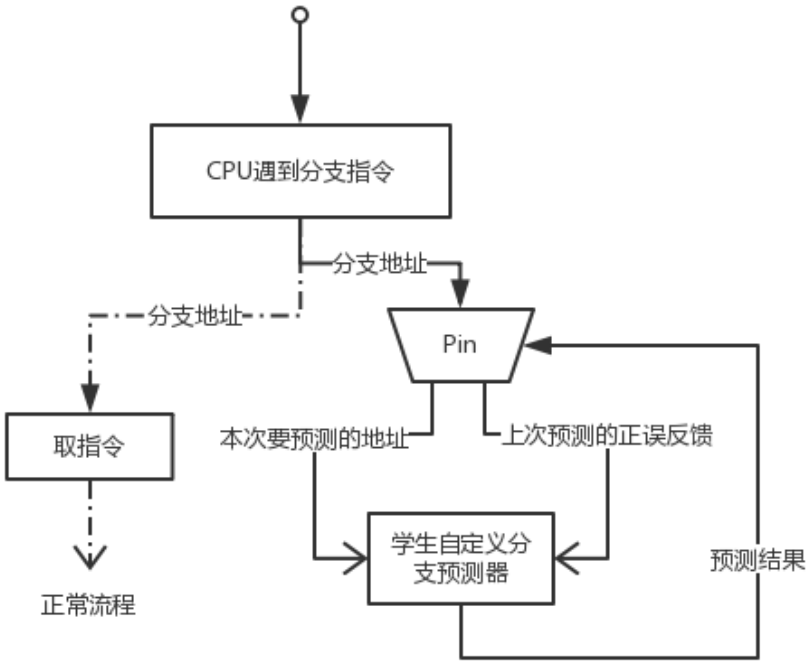
本实验的目的是探索分支预测器的体系结构。你的实现将扩展 BranchPredictor 类来实现你自己的分支预测器。从逻辑上讲，这个分支预测器将忽略处理器内核，而是直接接受来自处理器的分支地址查询。给定一个分支地址，分支预测器将返回一个表示分支是否被占用的回答。

BranchPredictor 类包含以下功能：

makePrediction：该函数接受分支指令的地址并产生预测。如果预测为跳转，该函数将返回 true，否则为 false。

makeUpdate：这个函数得到跳转结果反馈后更新预测表。它接受参数：分支地址，原始预测和分支的实际跳转方向。 你将使用此信息来更新你的内部分支预测器状态。

Pin 工具使用你的 BranchPredictor 实施如下。当工具遇到分支指令时，它会调用 makePrediction 来取得一次预测结果。然后通过与下一步实际执行地址比较后调用 makeUpdate 函数，记录这次比较结果。如果你使用的是动态分支预测，那么这个结果还可以被返回预测器用于改进你接下来的预测结果。原理图如下：



请注意!你的分支预测器只需要预测分支指令是否成功(即成功跳转 true,不跳转 false),不需要为它找到确切的分支目标地址。你的分支预测器允许使用不超过 33 KBytes 的持久性存储。使用超过 33 KBytes 将影响你预测器准确度的含金量。不用说的是，这对你的成绩也会有影响。你可以任意使用这些空间。在执行结束时，我们会跟踪这些信息。请在实验报告中也解释你使用的空间大小以及分配方法。

顺便提一下，与分支预测相关的计算机体系结构的子领域相当丰富，包含了许多重要的思想。尽管实施课堂上提出的分支预测可能也能取得不错的正确率，但如果你希望击败 TA 的参考分支预测指标，你可能需要自己做一些研究。提交时，请确保你的预测器的准确率至少在 90%以上!

这个实验是开放式的,最终的正确率可能还会与你的同学们进行比较。击败更多的对手,就能为你的本次实验获得额外的分数!

Report

请在实验报告中解释你预测器的整体工作原理，算法及流程，以及你为什么选择这种算法来预测分支的走向？如果你测试过好几种算法，那么也可以试着解释一下，为什么你选择的算法胜过了这些算法或是课堂上讨论过的算法？这可能会为你赢得额外的分数。具体详见实验报告模板。

Submit

请于实验发布的三周后提交本次实验要求的 BranchPredictor.h 和 lab.cpp，还有你的实验报告。

截止时间：2018.6.5　23:59

打包方式：115xxxxxxx_姓名_实验.zip

发送：邮件主题“115xxxxxxx_姓名_实验”，发送到 wchunpei@qq.com

Advice

我们的代码或基础架构中可能存在错误。如果你发现任何“有趣的”或“意外的”行为，这可能是我们提供的代码或基础架构中的问题。请立即将这些错误报告给我们。