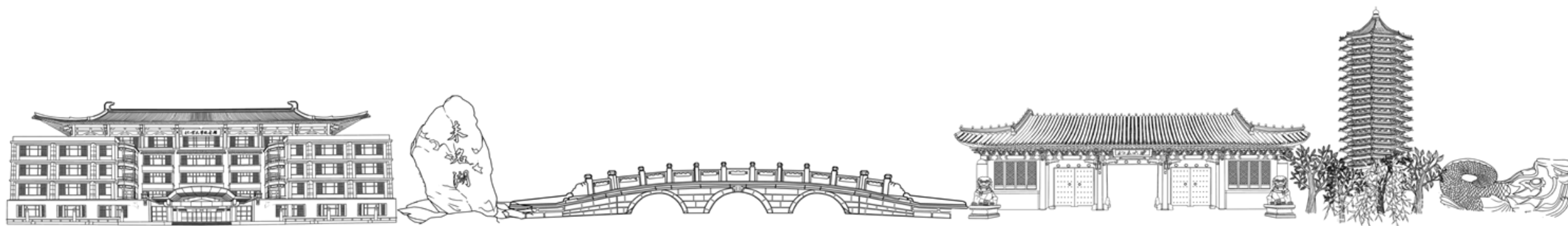


Combining branch predictors

汇报人：李蓉 时间：22.3.31





北京大学
PEKING UNIVERSITY

目 录

CONTENTS

01



摘要

02



介绍

03



各类预测器

04



总结

- 性能关键->指令级并行性的程度，瓶颈->条件转移指令
- 思路->硬件预测 不同预测器利用转移行为中不同模式
- 本文结合不同预测器的优点 动态指定最优预测器
- 提高转移历史有用性->与转移地址一起散列

优势

- 最大预测精度最优
- 给定预测器大小时也是最优的
- 给定的预测精度，硬件开销至少比其他方案小两倍
- 让单级历史数组的预测器优于具有多个历史级别的方案

- 超标量和超流水线技术提高指令级并行度
- 推测式执行流行加剧转移指令决定性能的重要，延迟转移（编译辅助）的有效性降低
- 转移性能问题的两个子问题
 - 预测转移方向
 - 转移目标指令最小延迟执行（BTB）
- 硬件转移预测策略
 - 双模转移预测（最著名）
 - 局部转移预测
 - 全局转移预测
 - 组合转移预测（本文）

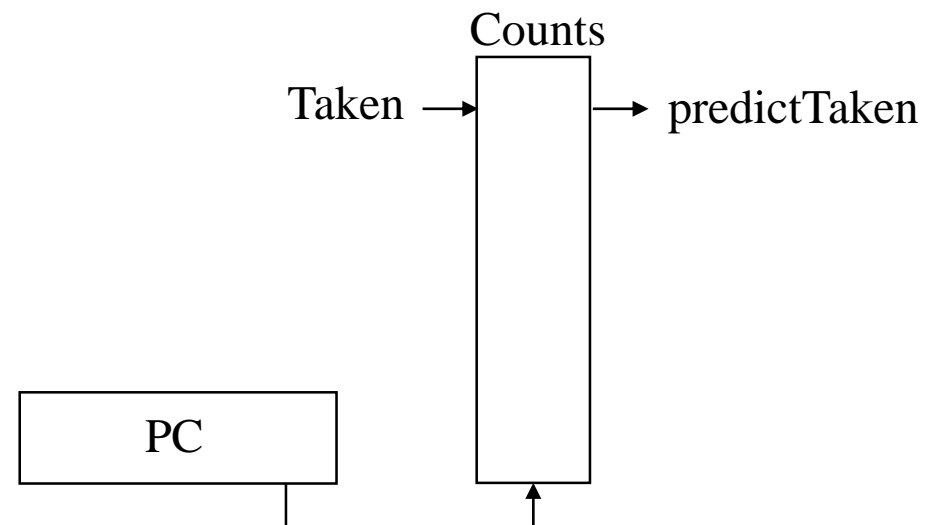
转移行为双峰分布

实现

- PC低位索引计数器表, 位宽2 bits
- 计数器加减是饱和的 (saturating)
- 最高有效位决定是否转移

不足及改进:

- 表数目小时, 多个转移共享, 预测精度降低
- 组相联查找为转移匹配带标记的计数器



局部转移预测

局部历史表: 每个条目是映射地址最近 n 个分支的转移方向

n 是条目的bit位宽, 计数器表以局部历史索引

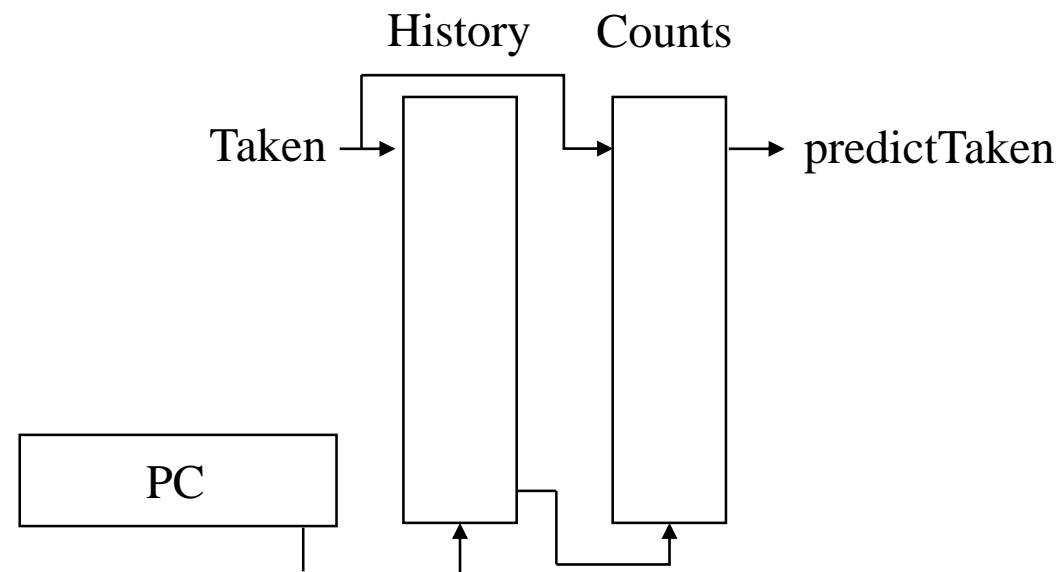
e.g. `for (i=1; i<=4; i++) { }`

- 模式 $(1110)^n$
- 历史位宽3 bits, 2^3 个计数器, 可全部预测准确。比如111时, 可推出下次不转移 (0), 011推出下次转移 (1)

不足:

- 所有映射到条目的转移局部历史混合在一起
- 计数器表被所有转移共享, 存在模式冲突

e.g. $(1110)^n$ 与 $(0110)^n$ 当转移历史是(110)



全局转移预测

全局历史:最近n个所有条件分支的转移方向

两种模式可发挥优势:

- 当前转移与最近转移强依赖

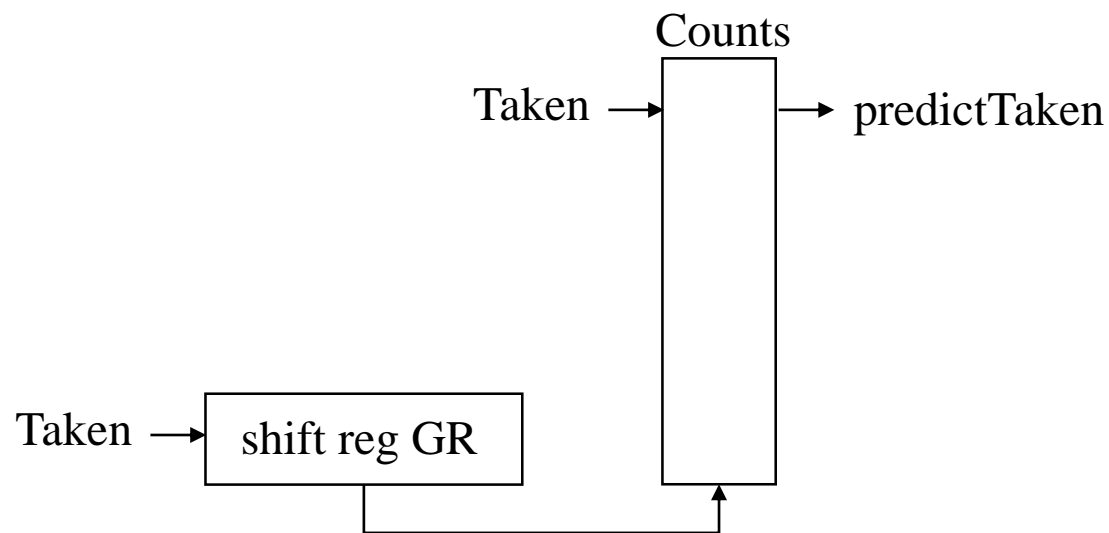
e.g. if (x<1) ...
 if (x>1) ...

- 全局转移预测可通过复制局部转移预测行为实现

e.g. for (i=0; i<100; i++)
 for (j=0; j<3; j++)

条件转移预测精度:

- 相同大小时, 全局方案低于局部方案
- 预测器小于1KB时, 全局方案信息含量小, 不如转移地址有效, 不如bimodal
- 每个转移90%时间朝同一方向走。预测器越大, 捕获信息越多, 更优于bimodal



索引全局转移预测

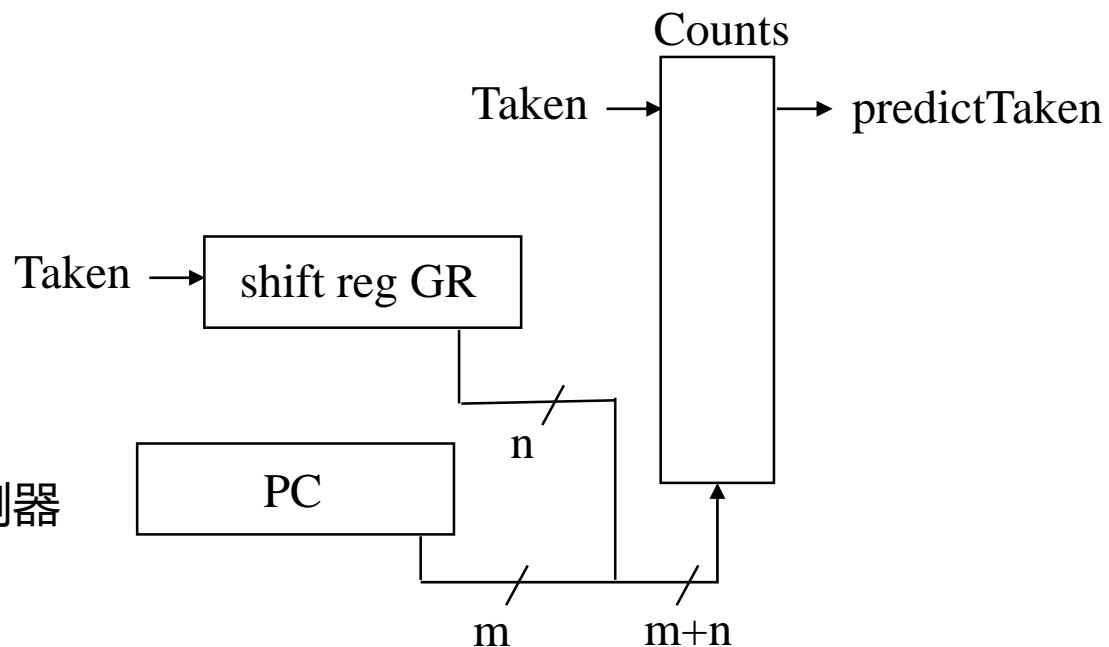
全局历史信息效率低于转移地址（识别当前转移）

实现

- 计数器表通过全局历史和转移地址位的串联进行索引
- 需要在使用更多历史位，还是更多地址位进行权衡

优势

- 基准测试中最佳精度预测器（gselect-best）在给定预测器大小时，优于双模和全局预测
- 小size时，gselect-best与双模预测的性能相当。一旦地址位足够，全局历史位越多，预测效果比双模越显著
- 小于1KB时，gselect-best优于局部预测，因为全局历史存储可忽略。更大size时，二者相近。更少数组访问，更低延迟，更易于流水线



全局历史与索引共享

gselect计数器索引存在冗余，地址位足够时，全局历史组合稀疏

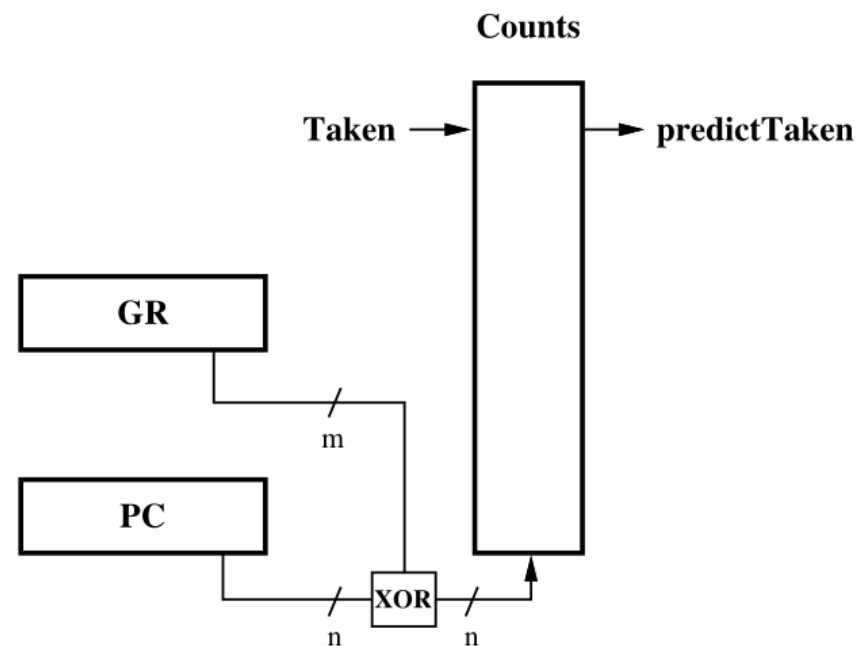
解决：将地址和历史组合作哈希；对地址和历史进行异或

e.g.

Branch Address	Global History	gselect 4/4	gshare 8/8
00000000	00000001	00000001	00000001
00000000	00000000	00000000	00000000
11111111	00000000	11110000	11111111
11111111	10000000	11110000	01111111

观察

- 只有gshare能区分所有四种情况，gselect无法利用高四位
- 特点
- 全局历史位可比转移地址位少
- 大于256B时，gshare-best略优于gselect-best
- 小size时，gshare的表现不如gselect



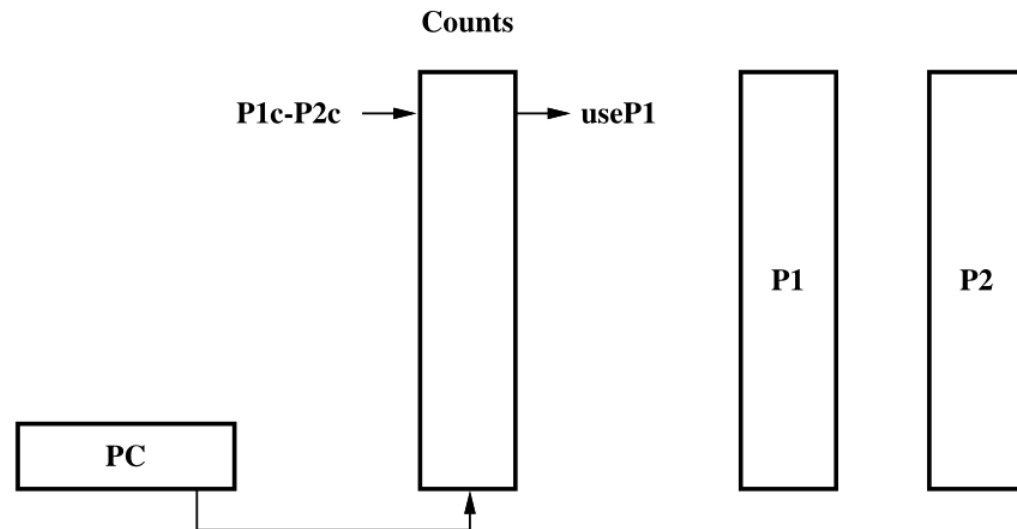
组合转移预测

结合不同转移预测器优势

实现

- 两个任意转移预测器P1和P2，选择最优预测器的计数器组
- 计数器2-bit上下饱和
- P1c, P2c表示P1和P2是否预测正确

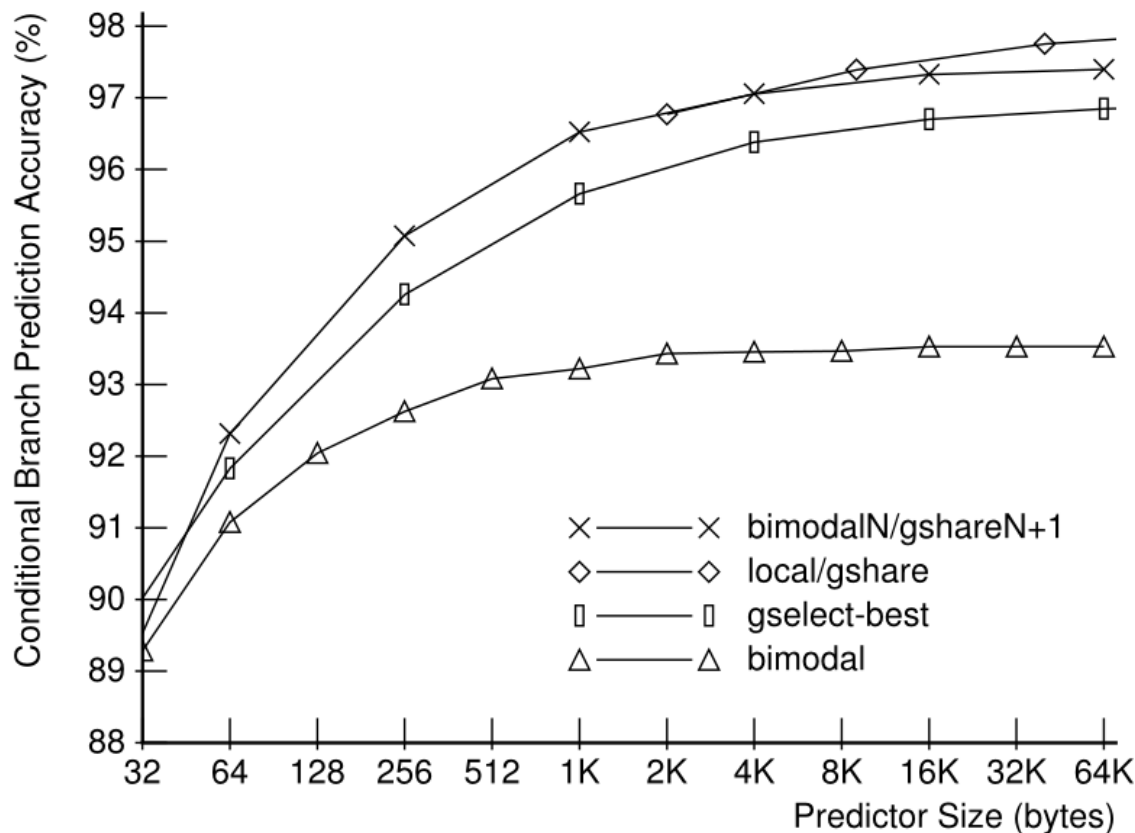
P1c	P2c	P1c-P2c	
0	0	0	(no change)
0	1	-1	(decrement counter)
1	0	1	(increment counter)
1	1	0	(no change)



组合转移预测

SPEC基准评估

- 组合预测器计数器数量是单独预测器的3倍，总是比单独预测器都好
- 组合中通常更频繁地使用双模预测器
- 一些不太可预测的基准，如gcc，组合预测器显著增加了错误预测分支之间的指令数
- bimodalN/gshareN+1的性能明显优于gselect-best，1KB前者接近16KB后者
- 大于2KB时，local/gshare精度优于bimodal/gshare。预测器更大时，该精度接近98.1%的正确率。因为大型局部预测器包含了双模预测器可用的信息



总结 ▶▶



01

使用全局分支历史和转移地址的位异或来访问计数器，对于给定的计数器数组大小，可获得更好的性能

02

通过跟踪哪个预测器更准确预测当前转移结合多预测器优势，以构建更精确的预测器

03

局部和全局的组合预测器的预测精度近98.1%，之前单预测器最精确方案为97.1%。随着利用指令级并行性，而错误预测转移将成为性能瓶颈的趋势增强，该方法愈有价值

谢谢

Thank You



北京大学
PEKING UNIVERSITY