



Combining Branch Predictors

Review By Group 11

一、引言

1. 介绍硬件分支技术的必要性和必然性。
2. 分支预测分类，可以分为两个子课题，第一是如何预测分支执行的方向，第二是如何使分支所跳转到的指令(分支目标指令)以尽量少的延迟被执行。一种快速提供目标指令的方法是用Branch Target Buffer(BTB),它是用于存放目标指令的专门指令缓冲，本论文着重于预测分支方向(第一个问题)。
基于硬件的各种分支预测策略：
3. 基于双模分支的预测技术(bimodal branch prediction)基于分支最后几次执行的方向，近年的工作显示使用跟多的分支历史可以更精确地预测；
4. 局部分支预测(local cranch prediction) 独立地考虑每个分支的历史，利用分支走向的重复性来预测。
5. 全局分支预测(global branch prediction) 使用最近所有的分支走向进行预测。

各有优点：

双模 —— 大部分分支偏向于一个特定方向

局部 —— 分支具备简单的重复性

全局 —— 一系列分支具备相互关联的走向关系

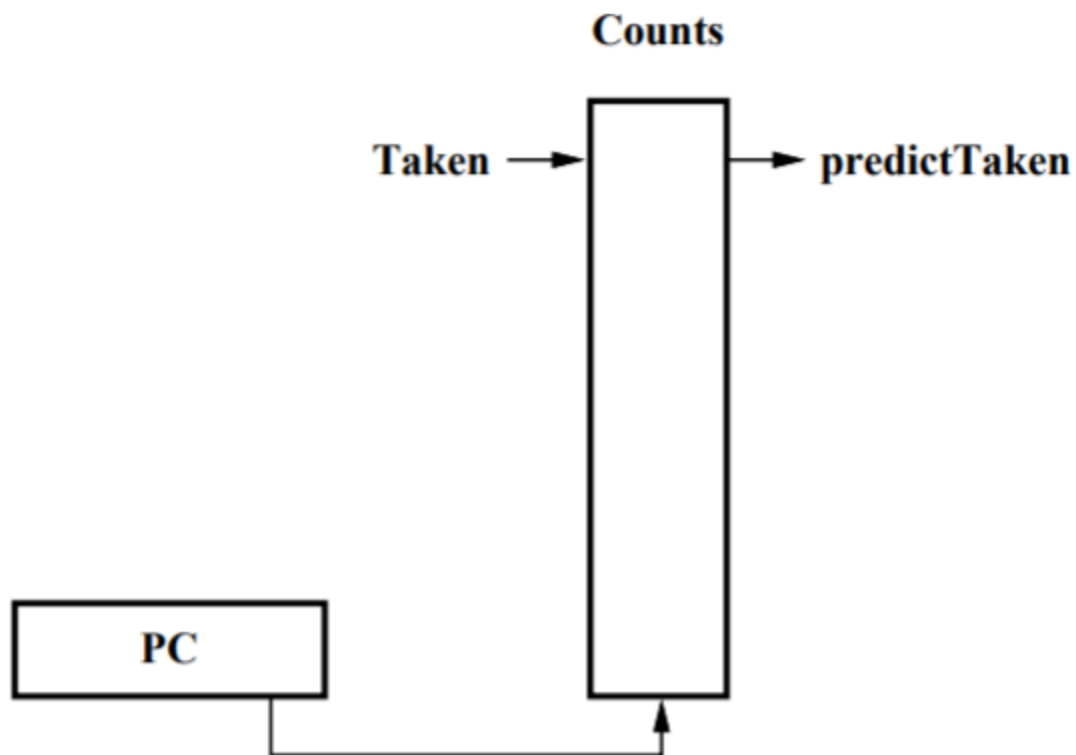
3. 本文介绍一种可以结合不同分支预测技术优点的方法。使用多种分支预测，然后对每次预测选择表现最好的方法。该方法效果比任何单种方法预测的效果更精确，文中也介绍了将分支历史与分支地址采用哈希表结构组合在一起以便检索的方式。

二、相关研究成果

三、双模、本地以及全局分支预测的量化评估结果

1. Bimodal Branch Prediction

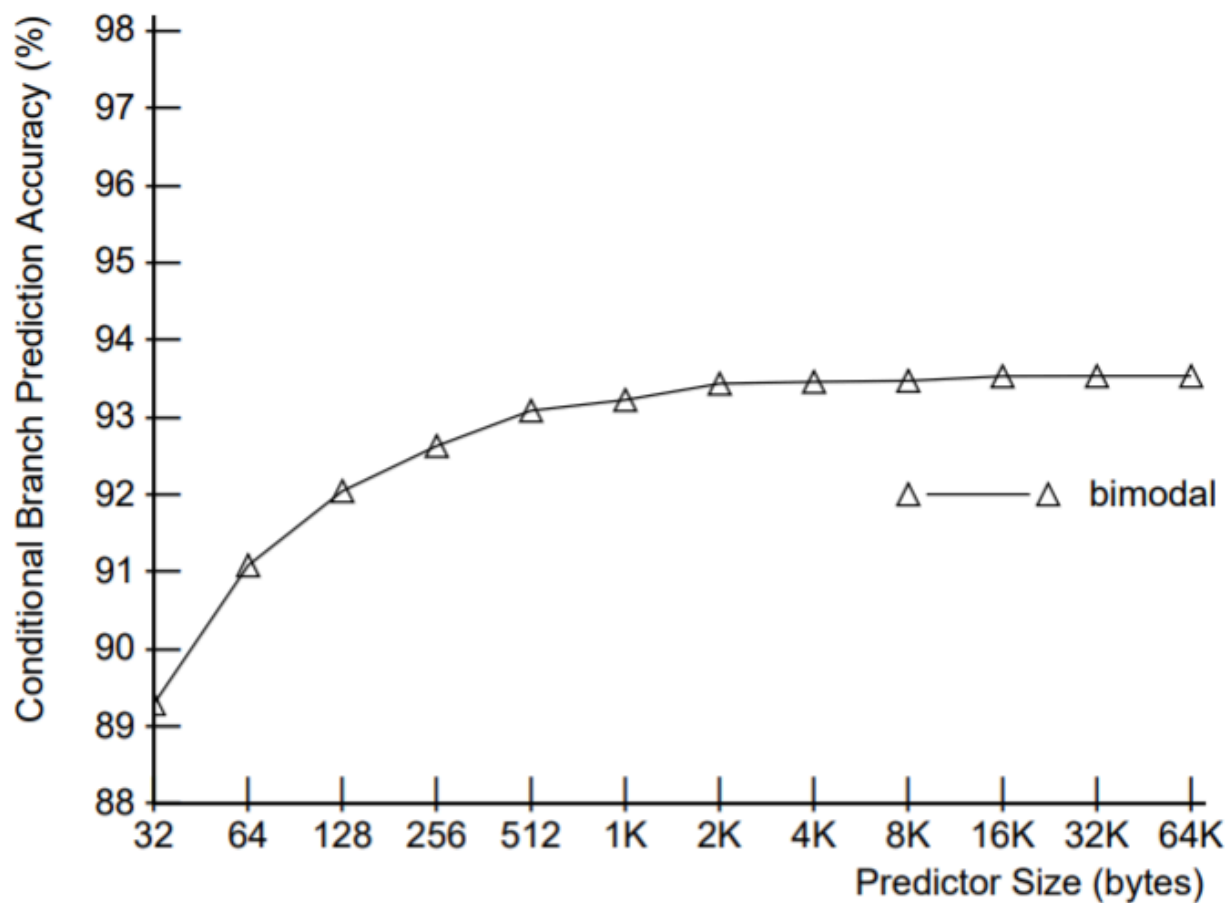
运用的大多数分支在执行过程中都具备的经常跳转或经常不跳转的特性(双模)。最简单的双模预测结构如图1所示，其包含一系列计数器所组成的计数器表，由指令计数器的地位地址进行索引。使用饱和寄存器设计的计数器(2bit长度)来记录，计数器对应指令跳转时，计数器=1;计数器对应指令不跳转时，计数器-1；分支预测的结果由计数器的最高位决定。



双模饱和计数器示意图



计数表足够大时，允许一条分支对应一个计数器表项；计数表较小时，多个分支使用同一个计数表项,性能表现会有所下降。可以为每位计数器设置标签(Tag)，采用多路联合的方法，把计数表项和分支指令对应在一起。



2. Local Branch Prediction

一种增加双模预测精度的方法，利用许多分支都具备可重复性。

其原理如下图所示，使用两个表格来进行分支预测。第一个表格记录当前分支的跳转历史(分支历史表branch history table)。每一个分支历史表项中，记录了根据指令低位地址索引到这个表项的那几条分支指令最近几次执行的方向，表项的长度与所记录的分支历史次数相同。第二个表依然采用我们前面介绍的双模饱和计数器表，有所不同的是，在这里双模计数器表项由分支历史表项索引。在本文中，由于分支历史表中记录的是单一分支的分支执行历史，因此将这种方案称为局部分支预测(local branch prediction)。

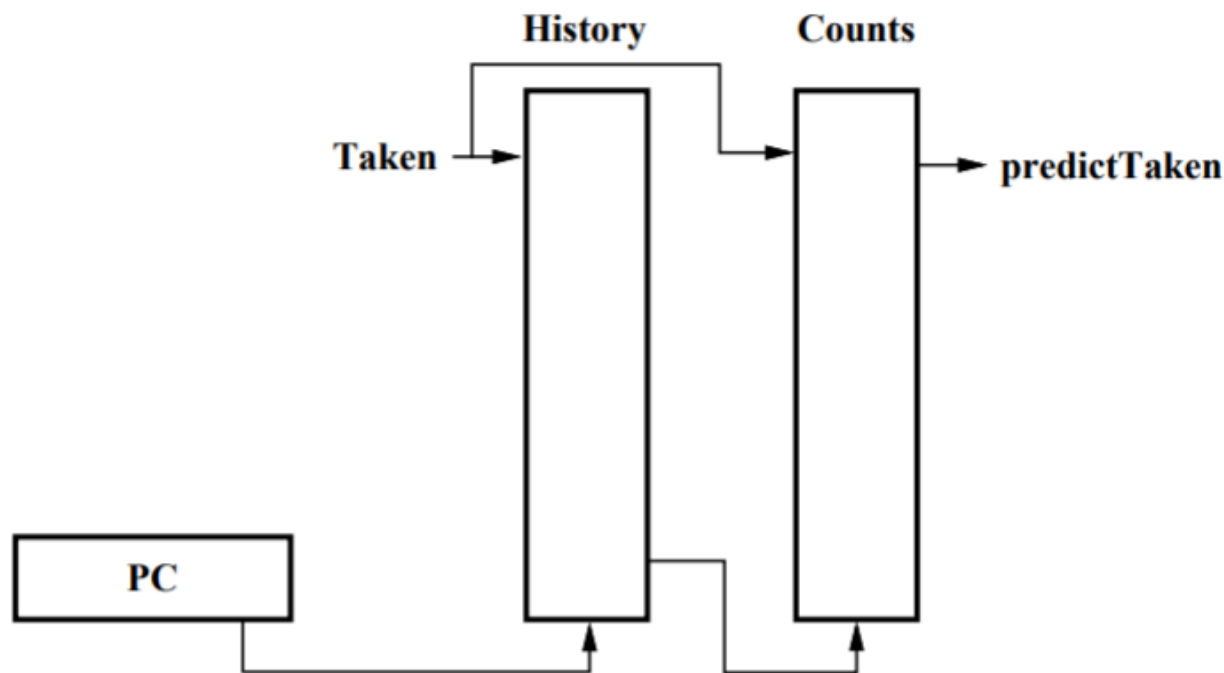
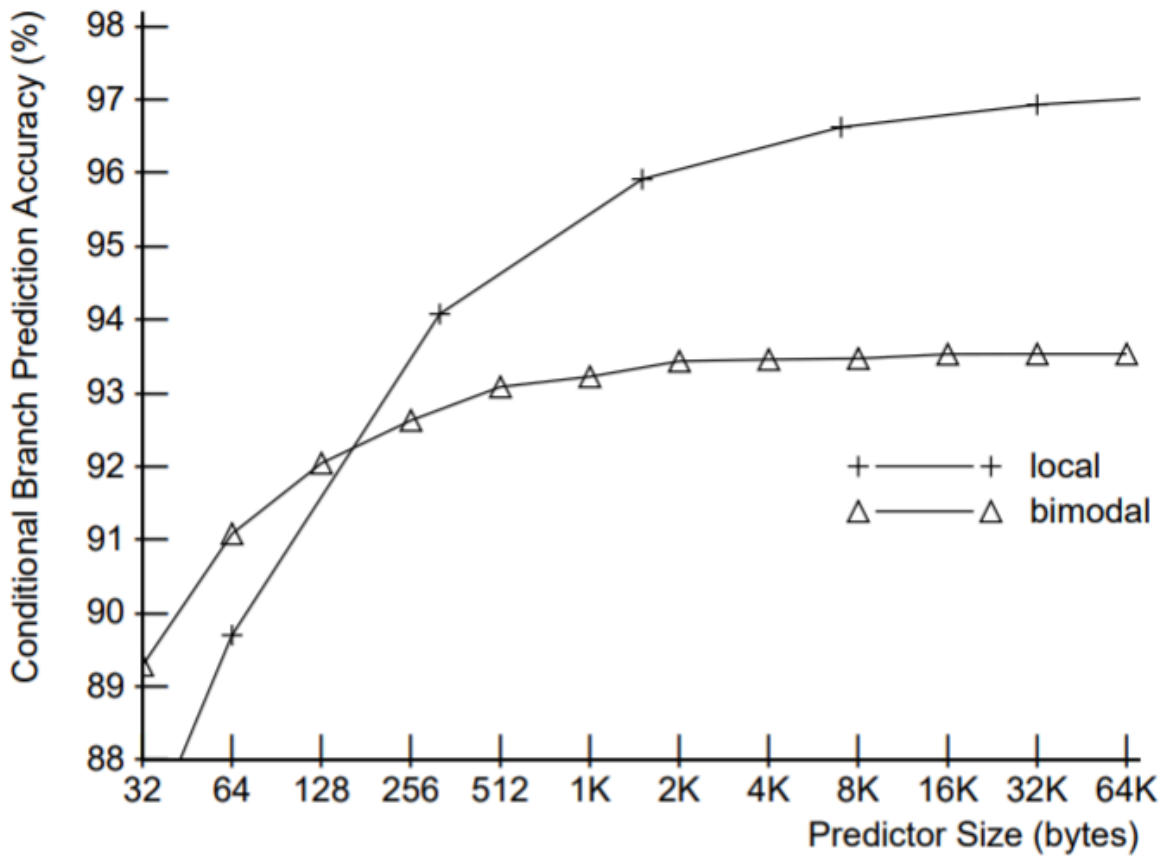


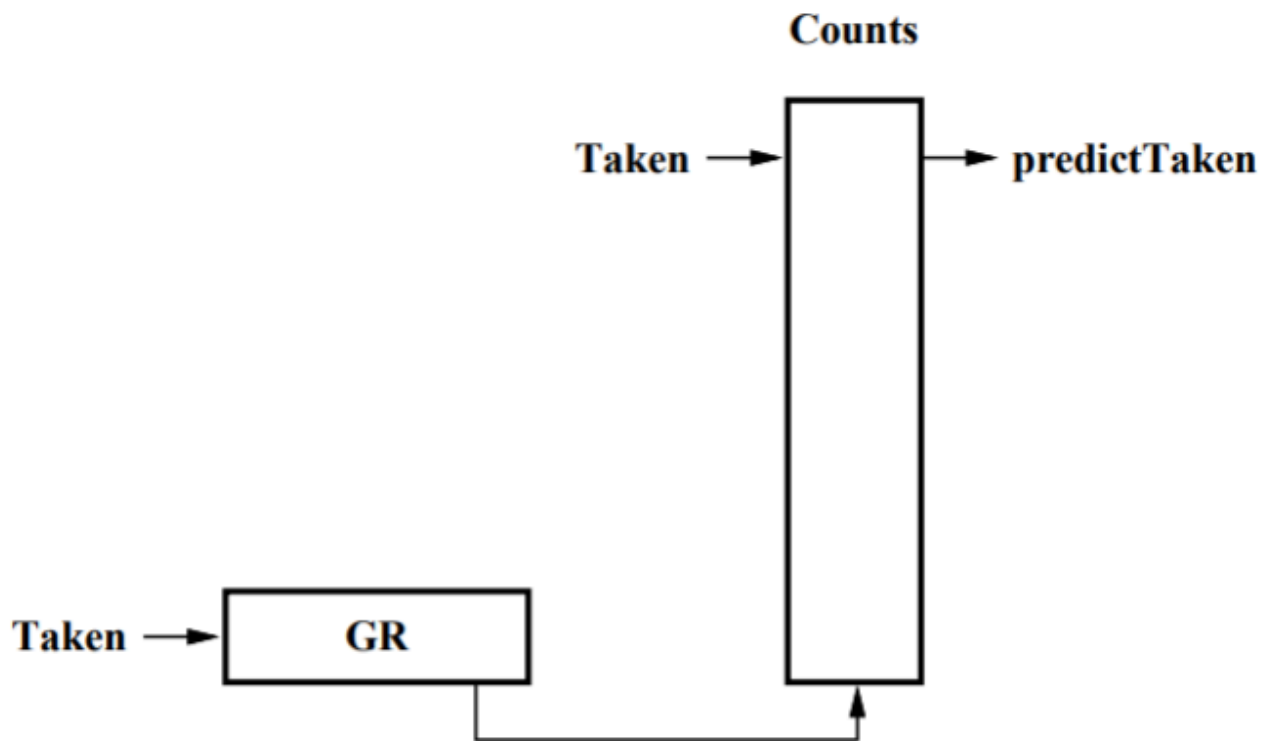
Figure 4: Local History Predictor Structure

下图显示了局部分支预测的性能曲线。简化起见，我们将分支历史表项与饱和计数器的表项设为相等。这种方案的更多其它变体我们将在附录A中讨论。表中可见，当预测器尺寸较小时，其性能表现甚至不如单一采用饱和计数器，因为此时预测器尺寸过小，共用同一个表项的分支指令较多，彼此之间相互干扰的情况也十分严重，可见如果预测器尺寸较小时，采用局部分支预测技术并没有太大意义。然而，当预测器尺寸增加到128字节以上时，局部分支预测器的性能表现将明显优于单独采用饱和计数器的性能。进一步增加预测器的尺寸，分支预测的精度还可以增加到97.1%左右，分支误预测率比单独采用双模预测时减小了一半以上。



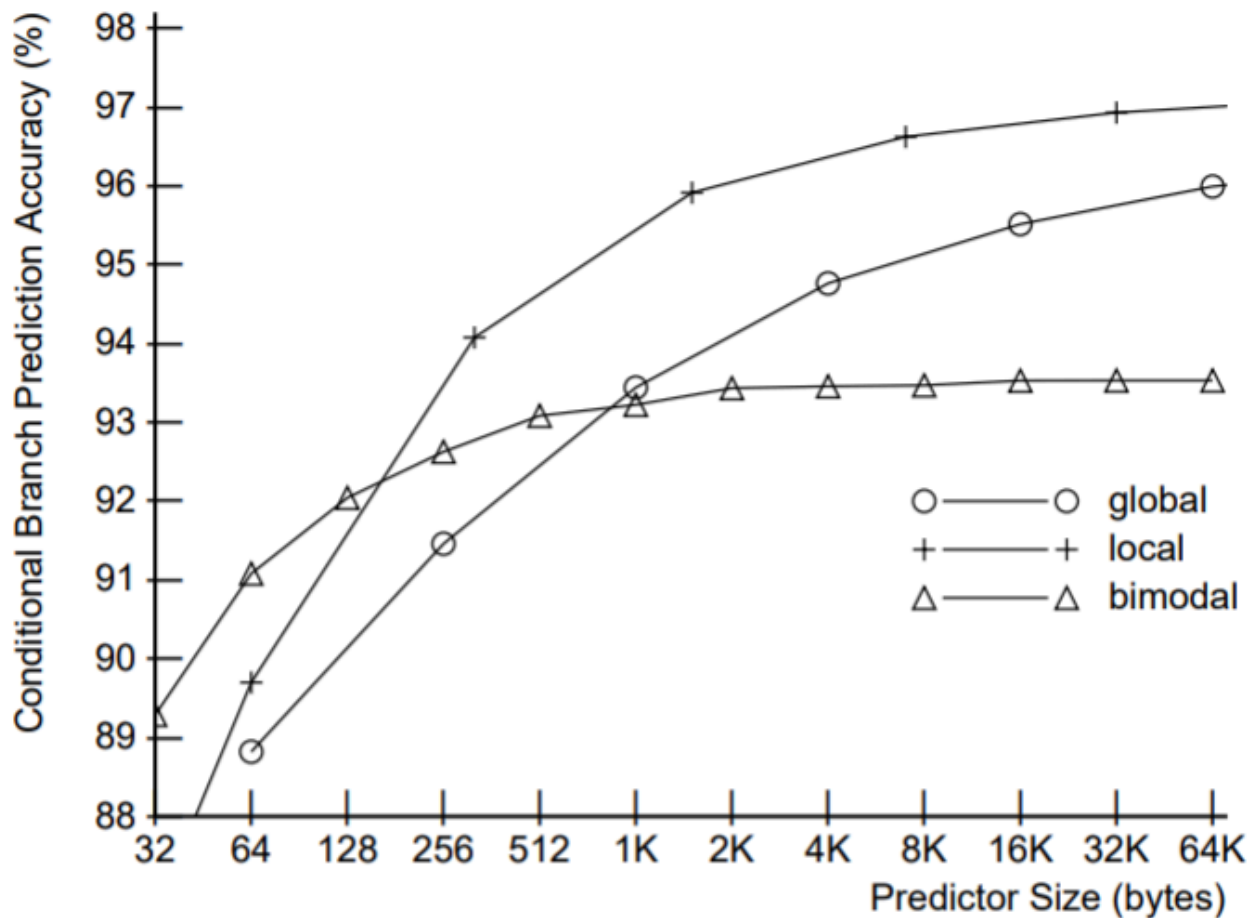
3. Global Branch Prediction

在前面介绍的局部分支预测技术中，我们只考虑当前单一分支语句的分支属性。Yeh和Patt又研究出一种可以根据其它分支语句的执行情况为当前分支做出预测的手段。这种技术的其中一种方案见下图：



这种方案中设置了一个移位寄存器，移位寄存器中存放了最近执行的几条分支指令的跳转情况。由于这种方法记录多条分支指令的跳转情况，因此又被称为全局分支预测。

全局分支预测方法具备两种优点。其一是当前分支语句的跳转方向与最近执行的几条分支指令存在紧密的相互关系时，全局分支预测可以充分发挥作用。第二种全局分支预测技术的优点是，它有时同样可以起到局部分支预测器的作用。当全局分支预测寄存器中能包含所有本地分支历史信息时，全局分支预测器可以达到和本地分支预测器一样的效果。

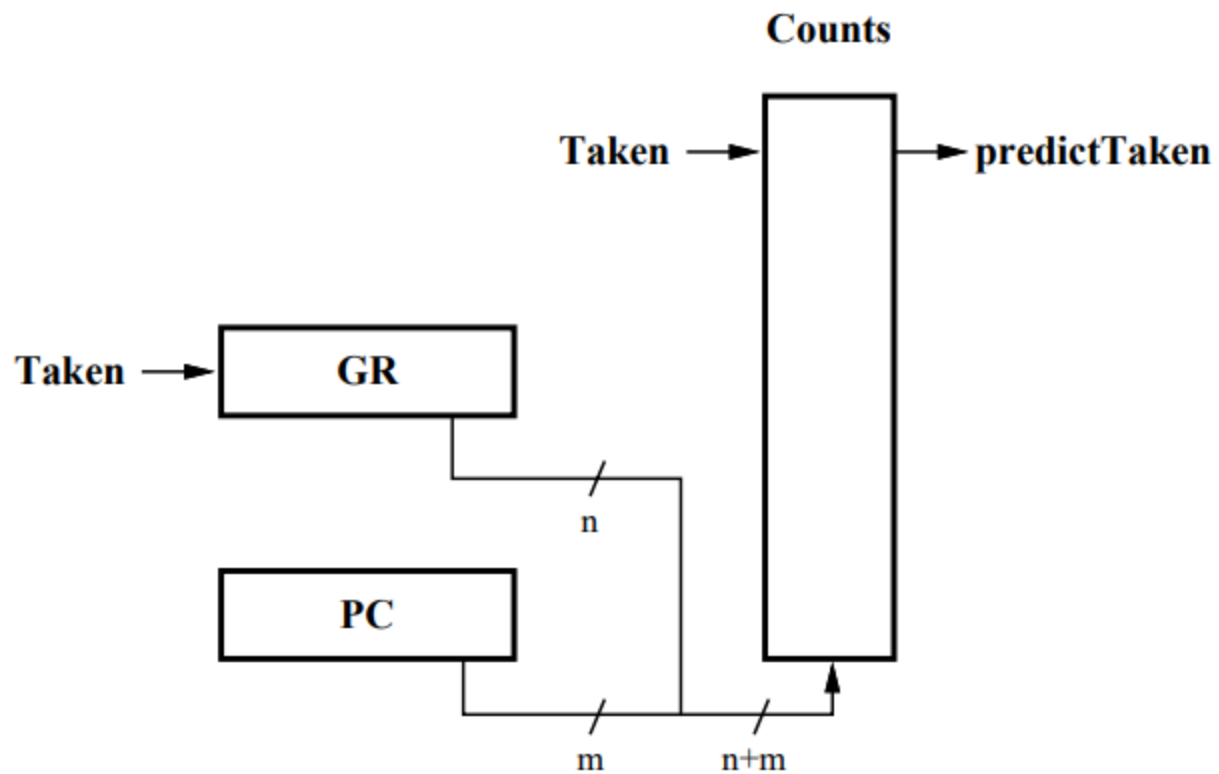


上图比较了全局分支预测、局部分支预测以及双模预测方式的性能曲线对比。图中可见，预测器大小相同时，全局分支预测器的表现并不如局部分支预测器好，而且只有在预测器容量达到1KB以上时，才比双模预测器好些。

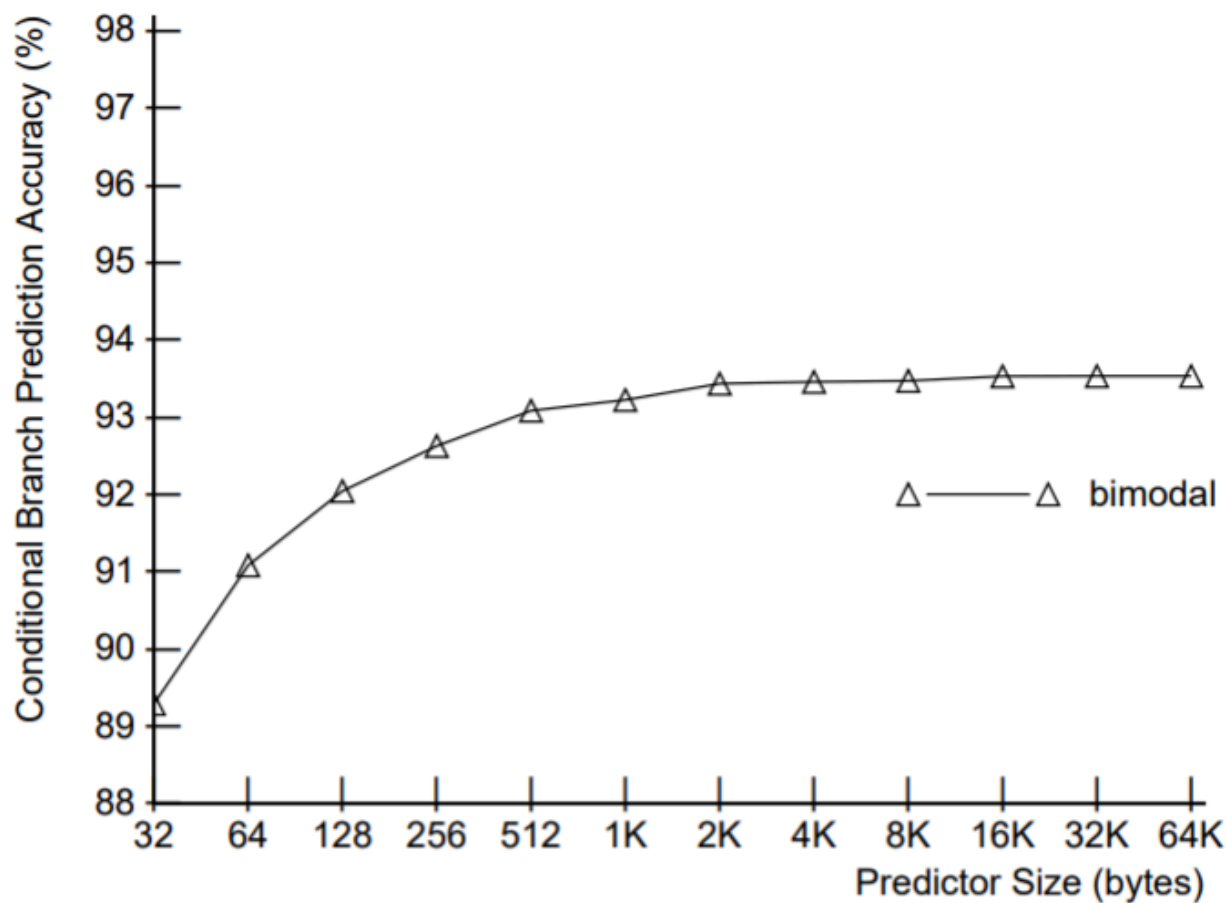
四、使用分支地址和全局分支历史建立哈希表检索分支预测结果的方法

1. 具有索引选择的全局预测器

全局历史信息在识别当前分支方面的效率比简单地使用分支地址要低，于是可以使用分支地址和全局历史进行预测，如下图所示，计数器表是用全局历史记录和分支地址位的串联索引的。



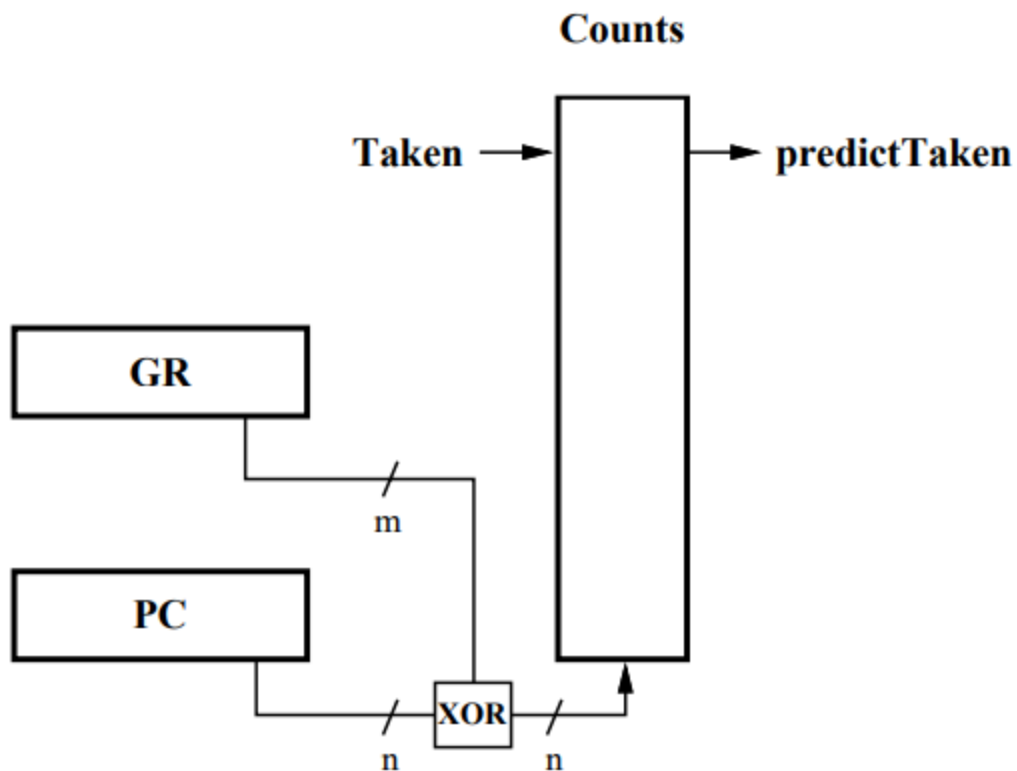
下图显示了使用选定地址位（gselect）进行全局预测的性能。使用位选择方法，可以在使用更多历史位或更多地址位之间进行权衡。图中并未显示所有这些可能性，而仅显示了给定大小的预测变量的性能，并且在整个基准范围内具有最高的准确性（gselect-best）。



2.具有索引共享的全局历史预测

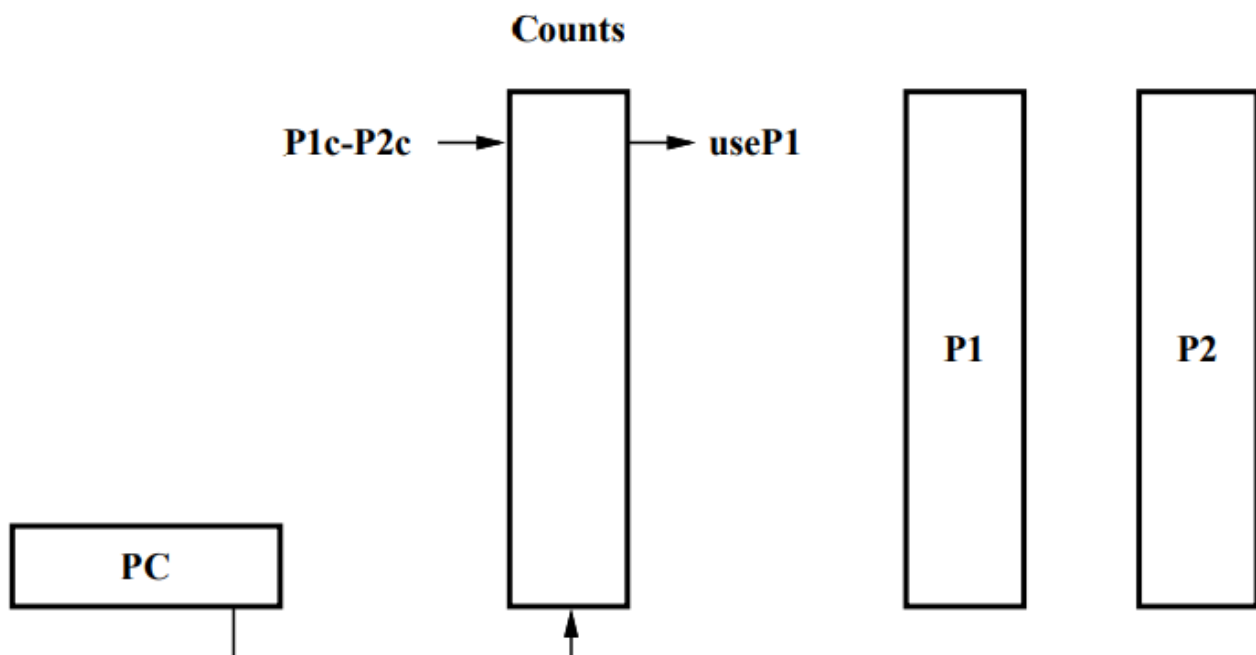
在对全局预测的讨论中，我们描述了全局历史信息如何弱识别当前分支。这表明gselect使用的计数器索引中有很多冗余。如果有足够的地址位来标识分支，则可以预期频繁的全局历史记录组合会比较稀疏。我们可以通过**将分支地址和全局历史记录哈希在一起**来利用这种影响。特别是，我们可以预期分支地址与全局历史记录的异或将比单独的任何一部分具有更多的信息。此外，由于使用了更多地址位和全局历史位，因此有理由期望比gselect更好的预测。

下图显示了gshare预测变量结构。

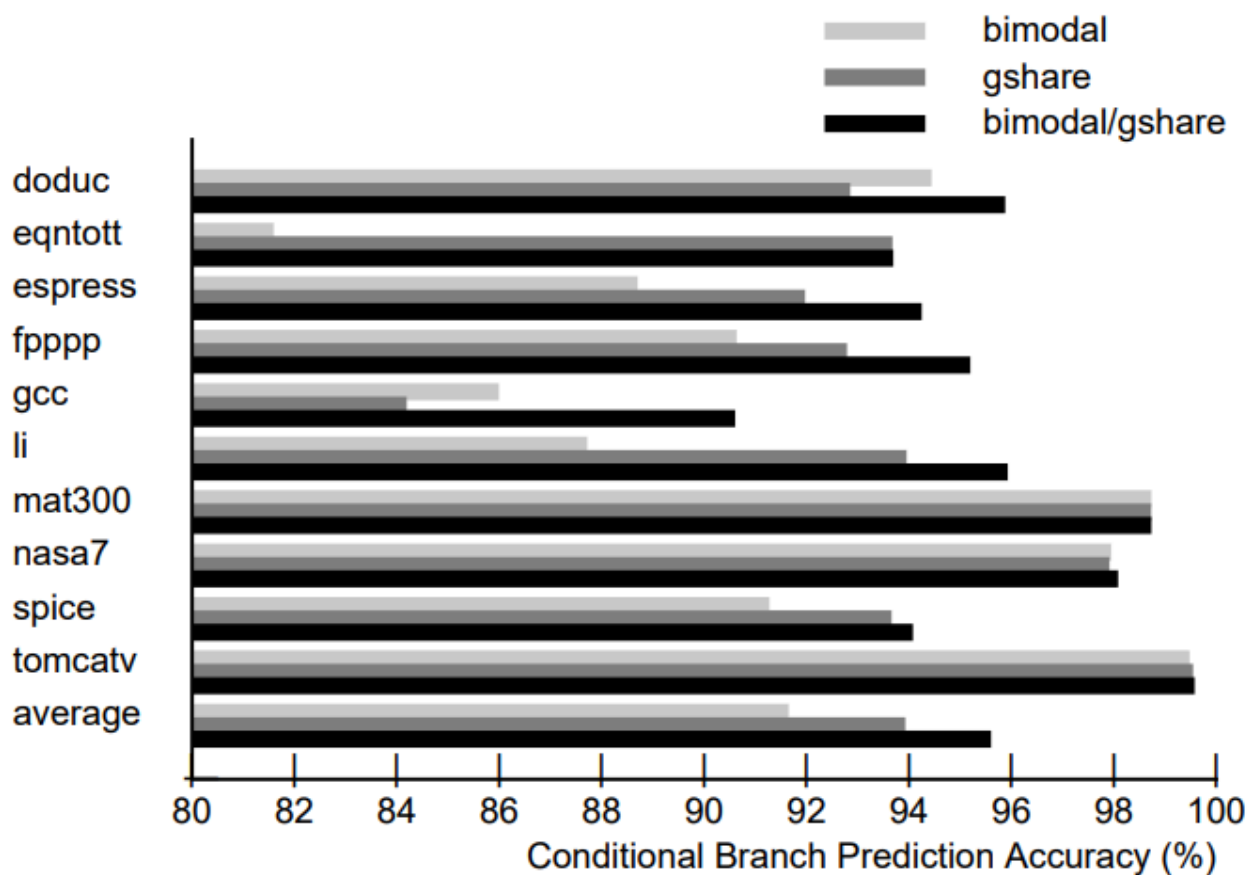


五、将各种预测技术混合在一起的预测技术

提出的不同分支预测方案具有不同的优势,一个自然的问题是, 不同的优势是否可以在具有更好的预测精度的新分支预测方法中组合。



如下图所示，组合的预测变量总是比任何一个单独的预测变量都要好。



六、总结

本文中提出了两种提高分支预测性能的新方法。

- 首先，证明了使用全局分支历史记录和分支地址的按位异或运算来访问预测变量计数器，可以在给定的计数器数组大小下获得更好的性能。
- 我们还表明，可以通过提供多个预测变量并跟踪哪个预测变量对于当前分支更准确来组合多个分支预测变量的优点。这些方法允许构造预测变量。对于给定的大小，这些方法比以前已知的方法更准确。同样，使用本地和全局分支信息的组合16个预测变量达到98.1%的预测准确性，而以前最准确的已知方案为97.1%。
- 随着机器设计人员尝试利用指令级并行性，并且误预测的分支成为关键的性能瓶颈，此处介绍的方法应会越来越有用。