

一种改进的神经网络分支预测技术

陈智勇, 廉海涛, 吴星星

(桂林电子科技大学 计算机科学与工程学院, 广西 桂林 541004)

摘 要: 通过研究神经网络的算法特征, 提出一种改进方法, 即设置一个门限值 η , 减小该算法的运算量以减小其访问延迟, 提高该算法的实用性. 研究表明该方法能够实现较快的访问速度.

关键词: 访问延时, 分支预测, 神经网络预测

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)11-0152-04

An Improved Branch Prediction Based on the Neural Network

CHEN Zhi-yong, LIAN Hai-tao, WU Xing-xing

(School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: By studying characteristics of neural network algorithm, this paper proposes an improved method, which sets a threshold value η to reduce the amount of calculation for reducing its access latency and improve the practicality of the algorithm. Studies show that this method can achieve faster access.

Key words: access latency; branch prediction; neural prediction

1 引言

目前大多数处理器均采用超标量超流水线结构来提高处理器的硬件资源利用率, 加快 CPU 的运行速率, 但是分支指令的存在却在一定程度上阻碍了超标量超流水线结构对处理器性能的提升. 流水线深度越深, 分支跳转失败带来的清空流水线的性能损失就会越严重. 而且随着现代编程中实现的功能越来越复杂, 分支指令的使用变得日益频繁, 其对流水线微处理器性能的影响也更加显著. 因此分支预测技术的应用对于减小流水线性能损失, 提高高性能处理器硬件资源利用率, 挖掘超标量超流水线结构的潜在性能起到至关重要的作用.

通过研究神经网络的算法特征发现, 在分支预测过程中, 虽然与当前分支相关性较弱的分支的权重值较小, 以至于不会影响到当前分支的跳转结果, 但是这些分支的权重依然会参与到当前分支跳转结果的运算当中, 从而增加了访问延时. 本文提出一种改进算法, 减少参与跳转结果运算的弱相关历史分支个数, 只提取强相关历史分支信息, 从而降低访问延时.

2 神经网络分支预测器

将神经网络引入分支预测领域, 许多新的分支预测机制被提出^[1-2]. 本文将针对传统神经网络分支预测算法提出改进.

2.1 传统神经网络算法原理

传统神经网络采用单层感知器构成一个人工神经元, 通过加权边界连接多个输入单元和一个输出单元. 每个感知器学习一个目标布尔函数 $t(x_0, \dots, x_n)$ 的 $n+1$ 个输入. 图 1 描述的是一个感知器模型. 感知器用一个权重向量表示, 输出值是权重向量 $(w_0 \dots w_n)$ 和输入向量 $(x_0, \dots, x_n)$ 的点乘 (其中 x_0 为 1, 作为偏移量输入). 感知器的输出 y 由下式计算得出.

$$y = w_0 + \sum_{i=1}^n x_i w_i \quad (1)$$

2.2 预测与更新机制

图 2 所示为传统神经网络预测器模型. 当处理器读取到一条分支指令时, 预测器根据分支指令地址哈希查找权重向量表, 得到当前分支所对应的权重向量 $(w_0 \dots w_n)$ (w_i 是一个有符号的八位二进制

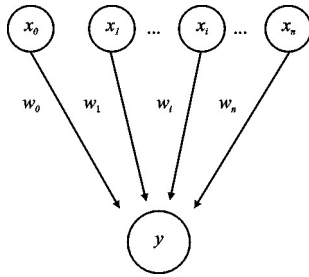


图1 感知器模型

数,其取值范围为 $[-128 \sim 127]$),其中 w_0 是当前分支的基准权重, w_1 至 w_n 分别对应于分支历史寄存器的第1到 n 位,即将分支历史寄存器中存储的分支历史作为输入向量。根据式(1),权重向量的每个权值同其对应的分支历史寄存器中存储的分支历史位相乘后做累加和,然后再加上基准权重得到预测结果 y 。 y 不小于零则表示当前分支指令的预测结果为跳转,否则表示该分支指令的预测结果不为跳转。

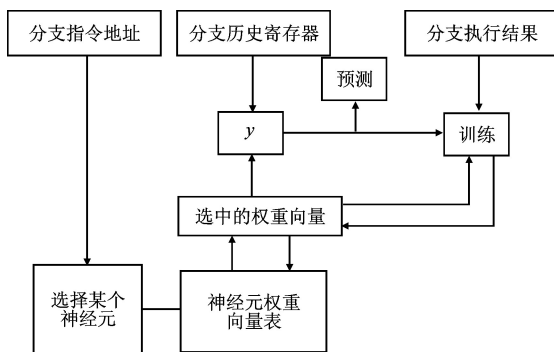


图2 传统神经网络预测器模型

例如,假设图中分支历史寄存器中保存的分支历史为 $(-1, 1, 1, -1)$;利用分支指令地址选择的权重向量为 $(1, 30, -2, -20)$ (权重的绝对值越高则表明其对应的分支与当前分支的相关性较强,反之则较弱。基准权重 $w_0 = 10$ 。那么根据式(1),最终的预测结果 $y = 10 - 1 + 30 - 2 + 20 = 57$,不小于零,则表示当前分支指令的预测结果为跳转。

当分支指令最终执行完时,预测器进入更新阶段。如果之前的预测结果的绝对值小于门限值 θ (用来判断预测器是否训练完成的一个参数^[3]),或者预测结果不等于真实结果,那么分支结果若为跳转,则基准权重加一,否则减一。同时令分支结果与 x_i 比较,相同则当前选中的权重向量中 x_i 对应的权重加一,否则减一。然后分支历史寄存器左移,存入当前分支结果。训练最终完成后,权重向量中与当前分支指令相关性较强的分支对应的权重值较高,相反与当前分支指令相关性较弱的分支对应的权重值

较小。

2.3 算法缺陷

虽然神经网络算法能够充分利用较长的分支历史信息得到相当显著的预测准确度,但是存在两个较为明显的缺陷:

第一,神经网络算法只有在充分的学习训练之后,才能得到较高的准确度,而往往学习训练的过程需要耗费相当长的时间。

第二,神经网络算法存在较高的访问延时。因为其在利用分支历史与权重向量点乘得到最终预测结果时,对于每条分支指令的预测都会进行 n 次乘法和 n 次加法(使用Wallace树也需要 $O(\log_{3/2} n)$ ^[4])运算。

现代处理器一般都采用多级流水线处理技术,对于第一个缺陷,常用的解决办法是采用多级组合分支预测算法,即将神经网络算法与Bimodal或者Gshare等算法相结合^[64],在神经网络充分学习训练之前,由Bimodal或者Gshare等算法预测分支结果。对于第二个缺陷,将在下一节中讨论。

3 基于传统神经网络算法的改进

3.1 算法改进

我们所运行的程序中有大量的分支指令,这些指令组合起来就表现为程序的分支行为,不同应用程序的分支行为非常复杂。分支指令之间并非是独立存在的,往往两条或者多条分支指令之间存在相关性,这种相关性通常表现为其中一条或者多条指令的跳转结果是其他临近的某一条或几条的分支跳转条件。当作为条件的分支指令执行完成时,与其临近的相关性较强的分支的执行结果就可以被精确的预测。以往的研究表明60%的分支指令具有相关性^[12]。以下是一个转移间相关的例子:

```
flag1=0;flag2=0;flag3=0
```

```
if (x<y)/* 分支 A */
```

```
flag1=1;
```

```
if(x<z)/* 分支 B */
```

```
flag2=1;
```

```
if(flag1&.&flag2)/* 分支 C */
```

```
输出显示 x 的值是最小的
```

```
if(y<z)/* 分支 D */
```

```
flag3=1;
```

```
if(flag2&.&flag3)/* 分支 E */
```

```
输出显示 z 的值是最大的
```

在代码段中有5条if语句,在编译后它们被翻

译成条件分支指令. 在这个例子中, 通过语义分析可以得到, 分支 A 和分支 B 的跳转结果直接影响分支 C 的跳转结果. 当分支 A 和 B 的跳转结果被确定, 分支 C 的结果也就被确定了. 由此可以确定分支 A 和 B 与分支 C 的关系属于强相关. 同理, 分支 B 和 D 与分支 E 的关系属于强相关. 而分支 A 和 B 的跳转结果并不影响分支 D 的跳转, 因此 A 和 B 同 D 的关系是弱相关. 同理, 分支 C 同 E 的关系是弱相关. 当处理器读取到分支 E 时, 分支历史寄存器中存储了 ABCD 的分支历史. 结合相关性与权重向量训练特点, 我们假设分支 E 对应的权重向量为 1, 30, 2, 20, 基准权重为 10, 如果分支历史为 $(-1, 1, -1, 1)$, 那么 $y = 10 - 1 + 30 - 2 + 20 = 57 > 0$, 减少弱相关分支历史后, $y = 10 + 30 + 20 = 60 > 0$. 如果分支历史为 $(1, 1, 1, 1)$, 那么 $y = 10 + 1 + 30 + 2 + 20 = 63 > 0$, 减少弱相关分支历史后, $y = 10 + 30 + 20 = 60 > 0$. 由此可以很容易发现, 当神经网络预测器在训练完成后, 如果处理器读取到分支指令, 预测器中与当前分支成弱相关关系的历史分支并不影响当前分支的预测结果, 但是它们却都参与了预测结果的计算, 因此就增加了预测器的额外访问延时. 而且随着记录的分支历史信息的增长, 弱相关分支个数也会随之增长, 这样就导致了访问延时的进一步增大.

针对这一问题, 本文提出一种改进方法——增加一组门限电路来剔除权重向量中弱相关历史分支对应的权重. 改进后的神经网络预测器结构如图 3 所示. 算法伪指令如下:

```
for i in 1...n do
  if  $|w_i| > \eta$  then
     $y = w_0 + x_i w_i$ ;
  end for
```

权重向量中, 除了基准权重外的每一个权重 w_i 对应一个门限电路, 当选中当前分支指令对应的权重向量后, 预测器将 w_i 作为输入送入其对应的门限电路中. 门限电路根据权重与设定的阈值 η 的比较结果, 选择输出原权重值或者输出 0. 然后门限电路的输出值与对应的分支历史做乘法后累加得到当前分支指令的预测结果. 例如对于上例, 设置门限电路组的阈值 $\eta = 5$. 在预测分支 E 时, 分支 A 和分支 C 的权重为 1 和 2, 在通过门限电路时, 分支 A 和分支 C 的权重的输出被屏蔽, 他们对应的门限电路输出为 0, 这样就将原本的四次乘法四次加法运算降低为两次乘法两次加法运算, 有效的降低了访问延时.

由于神经网络算法可以充分利用较长的分支历史信息, 因此合理设置阈值, 降低运算次数, 对提高预测器的访问速度是十分有效的.

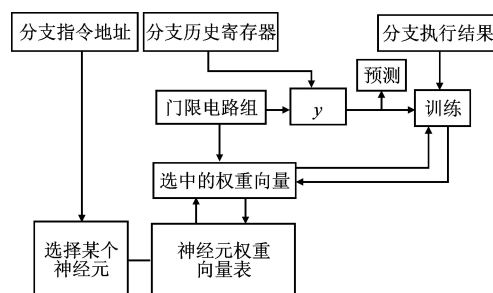


图 3 改进型神经网络预测器模型

3.2 模拟结果

我们采用 SimpleScalar 模拟器, 以 SPEC95 中 `fmath`、`math`、`printf` 三个基准程序作为目标测试程序. 在模拟测试中, 选择使用的分支历史寄存器长度为 8, 神经元数量为 1 024 个, 权重向量中每个权重的变化范围为 $[-128, +127]$, 总开销为 64 kB. 以分支预测准确率和总共运行的时钟周期数作为指标, 分别测试在传统神经网络预测算法及设定门限值 $\eta = 2, 8, 16$ 的情况下, 模拟器对测试程序的预测结果. 其中, 目标测试程序 `fmath` 运行 53 448 条指令, `math` 运行 213 688 条指令, `printf` 运行 1 812 877 条指令.

模拟器的预测准确率如图 4 所示, 图中 $\eta = 0$ 代表的是传统神经网络预测器(即不对预测器添加门限电路)对三个基准测试程序的预测正确率, $\eta = 2, 8, 16$ 分别代表三种不同门限条件下, 改进型预测器的预测准确率. 从图中可以看出, 当执行只有 5 万条指令的 `fmath` 程序时, 设置门限会降低预测器的准确率, 而执行指令数较多的 `math` 和 `printf` 程序时, 设置门限则能够提高准确率. 而且随着程序指令数的增多, 预测器准确率提高的比例也会逐步加大.

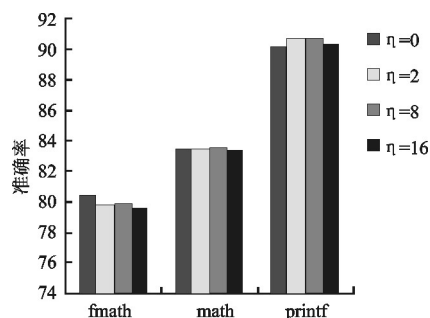


图 4 预测准确率

模拟器运行目标程序总的时钟周期数如表 1 所

示.可以发现,在节省预测周期方面,对于 fmath 程序,传统神经网络预测器要优于改进型预测器,对于 math 程序,两者的预测时长相当,而在执行 printf 程序时,改进型预测器则明显优于传统神经网络预测器.

表 1 模拟器执行总时钟周期数

	$\eta=0$	$\eta=2$	$\eta=8$	$\eta=16$
fmath	79 811	80 501	80 362	80 583
math	263 448	263 696	263 239	263 724
printf	1 882 832	1 870 279	1 868 046	1 874 161

综合以上实验结果,我们得到当执行的指令数达到 20 万条以上时,改进的神经网络预测器能够在保证不降低预测准确度的前提下降低分支预测延迟.而且,对于分支历史寄存器长度为 8,权重范围为 $[-128, +127]$ 的神经网络预测器,当门限值设定为 $\eta=8$ 时,具有更好的预测效果.

3.3 性能分析

基于传统神经网络的改进算法与其他算法相比具有很多优势.首先,同两级自适应预测算法相比,该改进算法拥有可以利用较长分支历史而不会引起硬件开销指数级增长的优势.虽然在训练未完成之前,预测准确度不高,但是可以通过结合其他预测算法进行弥补.其次相比传统神经网络预测算法,该改进算法减小了预测访问延时,提高了传统神经网络的实用性和可实现性.第三,与丹尼尔·A.希梅内斯(Daniel A. Jimenez)提出的 Path-Based 分支预测算法^[7]和分段线性分支预测算法(Piecewise Linear Branch Prediction)^[2]相比,虽然 Path-Based 分支预测算法在访问延时和预测准确度方面更具优势,而分段线性分支预测算法主要对线性不可分分支具有更好的预测性能,但是这两种算法空间复杂度都非常高,而本文提出的改进算法则更容易实现.相比增加空间复杂度来减小访问延时提高预测准确度,传统神经网络算法做出改进,实现起来更为合理.

4 结束语

本文通过研究神经网络预测算法特征,发现分支历史中存在的弱相关分支指令对当前分支的预测不会产生太大影响,但是却会增加额外的预测器访问延时,针对这一问题,提出增加门限电路的方法,能够在不降低原预测器的预测准确度的基础上,提高访问速度.

目前分支预测技术发展遇到了一个瓶颈,其最高精度达到 95%~98%,很难进一步提高.但是随

着大量新应用程序的出现,又有新的程序分支行为,以及对硬件开销、功耗等因素考虑,分支预测技术将会有一些新的发展.

参考文献:

- [1] Daniel A Jiménez. Fast path-based neural branch prediction[C] // Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture Piscataway, NJ: IEEE, 2003.
- [2] Daniel A Jiménez. Piecewise linear branch prediction[J]. Computer Architecture News, 2005, 33(2): 382-393.
- [3] Daniel A Jiménez, Calvin Lin. Neural methods for dynamic branch prediction[J]. ACM Transactions on Computer Systems, 2002, 20(4): 369 - 397.
- [4] Daniel A Jiménez, Calvin Lin. Dynamic branch prediction with perceptrons[C] // Proceedings of the Seventh International Symposium on High Performance Computer Architecture. Austin, Tx: IEEE, 2001: 197 - 206.
- [5] Daniel A Jiménez. Generalizing neural branch prediction[J]. ACM Transactions on Architecture and Code Optimization, 2009, 5(4): 52-61.
- [6] Gabriel H Loh, Daniel A Jiménez. Modulo path history for the reduction of pipeline overheads in path-based neural branch predictors[J]. International Journal of Parallel Programming, 2008, 36(2): 267 - 286.

作者简介:



陈智勇 男,(1971-),副教授,硕士研究生导师.研究方向为计算机体系结构、嵌入式系统.



廉海涛 男,(1986-),硕士.研究方向为计算机体系结构.