

微处理器

转移预测部件

流水线

设计

②

6-8

现代微处理器转移预测设计策略

Design Policy of Branch Prediction for Modern Microprocessors

张建林

王昭顺

TP332.02

(首都师范大学计算机系 北京 100037) (北京科技大学计算机系 北京 100083)

Abstract Branch instruction (especially conditional branch instruction) is a key of effecting performance of modern microprocessors. To reduce penalty brought by branch instruction as far as possible is a hotspot for researching high-performance microprocessors. In this paper, we discuss the effect of efficiency brought by conditional branch instruction and concerned processing technology, and analysis the design policy of branch prediction for modern microprocessors.

Keywords Microprocessor, Branch prediction, Processing technology, Design policy

现代微处理器普遍采用流水线、超级流水线、超标量或 VLIW 等技术来提高指令级并行度 ILP,但也带来由于条件转移指令导致流水线断流的效率损失问题,现代微处理器一般采用转移预测部件来尽量减小这种影响。

一、条件转移对流水线的影响

指令重叠执行是现代微处理器提高 ILP 的途径。影响指令重叠执行的主要因素是因转移指令引起的控制相关。无条件转移指令对程序执行速度的影响很小,而条件转移指令在转移成功时对程序执行速度的影响很大。为了解决条件转移对流水线的影响,通常采用“猜测法”,即在条件转移指令之后选择一个分支方向,让后续指令进入流水线执行。当分支方向猜测正确时,流水线的效率没有损失,但当分支方向猜测错误时,流水线中有多个功能段要被浪费,使流水线吞吐率下降的百分比为:

$$D = pq(k-1) / [1 - pq(k-1)]$$

其中 p 是条件转移指令在程序中所占的比例, q 是转移成功的概率, k 是流水线的功能段数。根据统计,一般程序中条件转移指令占指令总数的五分之一左右 ($p = 20\%$),当转移成功的概率为 $q = 60\%$,流水线有 8 级时,流水线的吞吐率将下降 46%^[1]。

二、条件转移的处理技术

由于条件转移指令对流水线,特别是超级流水线的影响非常大,必须采取措施来减小这种影响。现代微处理器对条件转移的处理技术有:

1) 延迟转移技术和指令取消技术。由于转移指令

有两周期的转移延迟,在遇到转移指令时,依靠编译器把一条或几条没有数据相关和控制相关的指令调度到转移指令的后面。当被调度的指令执行完后转移指令的有效目标地址已经计算出来。

在使用指令取消技术的处理器中,所有转移指令和数据变换指令都可以决定下面待执行的指令是否取消。采用的规则是:向后转移在转移成功时取消下条指令,否则执行下条指令;向前转移在转移不成功时执行下条指令,否则取消下条指令。

2) 静态转移预测技术。按固定方向(转移成功或转移不成功)预测转移方向。静态转移预测可以用软件,也可以用硬件实现,还可以在转移的两个方向上都预取指令,这时硬件要设置一个转移目标缓冲区 BTB。

3) 动态转移预测技术。根据近期转移的历史记录预测下一次转移的方向,可以随程序的执行动态改变转移的预测方向。实现动态转移预测有三种方法:

① 转移预测位。在指令 Cache 中设置“转移历史表(BHT)”字段。在执行转移指令时,把转移成功与否的信息记录在 BHT 中,当下次有这条转移指令时,转移预测逻辑根据 BHT 中记录的信息预测转移是否成功(转移的方向)。采用这种技术时,需要不断修改 BHT,修改 BHT 的规则和预测结果用 BHT 的状态转换图(BTB)表示。

② 转移目标地址 Cache(BTAC)技术。用转移目标地址 Cache 来保存最近几次转移的 BHT 和转移目标地址。转移目标地址 Cache 如图 1。

当程序执行到一条转移指令时,把当前指令地址与转移目标地址 Cache 中的所有转移指令地址比较,如果发现相等,则根据转移历史表预测本次转移方向,

同时用转移目标地址预取指令。

转移指令地址	转移历史表	转移目标地址
I_0	H_0	P_0
I_1	H_1	P_1
I_2	H_2	P_2
...	...	...
I_n	H_n	P_n

图 1 转移目标地址 Cache

③转移目标指令 Cache(BTIC)技术。用转移目标指令 Cache 来保存最近几次转移的 BHT 和转移目标之后的几条指令。转移目标指令 Cache 如图 2。

转移指令地址	转移历史表	转移目标地址之后的 m 条指令				
A_0	H_0	$I_{0,0}$	$I_{0,1}$	...	$I_{0,m-1}$	
A_1	H_1	$I_{1,0}$	$I_{1,1}$	...	$I_{1,m-1}$	
A_2	H_2	$I_{2,0}$	$I_{2,1}$	...	$I_{2,m-1}$	
...	...	...	...	...	...	
A_n	H_n	$I_{n,0}$	$I_{n,1}$	...	$I_{n,m-1}$	

图 2 转移目标指令 Cache

当程序执行到一条转移指令时,把当前指令地址与转移目标地址 Cache 中的所有转移指令地址比较,如果发现相等,则根据转移历史表预测本次转移方向,若转移成功,预取转移目标地址之后的指令。

4)提前形成条件码技术。条件转移指令造成流水线吞吐率下降的主要原因是条件码形成得太晚。一般条件转移指令的条件码由上一条运算指令产生,而大多数情况下条件码可以在运算开始之前或运算过程中就能产生,因此在硬件设计中,尽量不要在运算完成之后再形成条件码,此外对编译程序设计,也应尽量将产生条件码的指令调度到转移指令之前几个周期。

三、现代微处理器的转移预测部件设计策略

90 年代微处理器市场形成五种 RISC 芯片与 Pentium 系列芯片竞争的局面。这些微处理器的转移预测部件设计策略如下:

● PowerPC620 设置了 2048 个转移历史表(BHT)和 256 的转移目标地址 Cache(BTAC),采用改名缓冲器技术实现动态转移预测,最多可嵌套预测 4 层转移指令,并推测执行相应路径上的指令。

● UltraSPARC 是第一个 64 位 SPARC 系列微处理器,允许指令乱序执行,设置了 2048 个转移历史表(BHT),并在指令 Cache 中设置特殊字段作为转移

目标的预判机构,实现单周期转移追踪技术和动态转移预测。

● PA-RISC7200 采用静态转移预测。

● MIPS R10000 允许指令乱序执行,设置了 2048 个转移历史表(BHT)和 64 个全相联 TLB,采用动态寄存器改名技术实现动态转移预测,能够推测执行 4 层转移预测路径上的指令。

● Alpha 21164 采用静态转移预测,根据 2 个历史记录位对所有转移指令进行预测。对以前未执行过的分支 Alpha 使用位移符号确定预测方向,对于以前已执行过的驻留于指令 Cache 中的分支指令,Alpha 使用历史记录位加以标识,这样简化了指令的流程控制,保证能通过提高时钟来增加性能。

● Pentium 采用动态转移预测,设置了两个 32 字节分支目标缓冲器(BTB),一个预取顺序执行的指令,直到取到一条分支指令,预测转移是否发生,如果预测转移不发生,指令预取继续进行,若预测将进行转移,则第二个预取目标缓冲器根据这一预测开始预取指令。

微处理器	延迟转移	静态转移预测	动态转移预测	提前形成条件码
PowerPC620			BTAC	
UltraSPARC			BTIC	
PA-RISC7200				
MIPS R10000			BTAC	
Alpha21164				
Pentium 系列			BTIC	

通过以上分析,现代微处理器中已不用延迟转移技术作为转移处理策略。延迟转移技术适用于流水线级数不大的单流水线标量处理器。现代微处理器的流水线的级数较大,需要调度到转移指令后的指令数较大,而调度两条指令以上的成功不高,在转移指令之后仍需插入若干个空周期,从而影响微处理器的效率。

提前形成条件码技术是一种理想的转移处理策略,但同样依赖于编译器对指令的调度,当调度成功不高时,不能消除转移的影响。

静态转移预测机制简单、实现代价小,但灵活性差,不适应程序的实际情况。在采用简单的指令调度方法与流水线,通过提高时钟频率来提高微处理器性能的微处理器设计策略时(如 Alpha 21164)采用静态转移预测技术。

动态转移预测因其相对最高的性能得到广泛应用,但其预测机制较复杂,实现代价较大,在利用更灵巧、更复杂的指令调度技术和流水线技术来提高微处

8-10 A-NET 模拟程序中进程间通信接口的实现^{*}

The Implementation of the Communication Interface between the Processes of the A-NET Simulator

吕英华

TP316

(东北师范大学计算机科学系 长春 130024)

Abstract This paper describes how to utilize the functions of UNIX process management and process communication to form the communication channel between the processes of a A-NET simulator, the semaphore for holding communication channel, and the reading request signal. It also discusses the communicating realizable method

Keywords Pipe, Communication channel, Semaphore, Reading request signal

1 引言

A-NET 计算机是专用于执行面向并行对象语言 A-NETL 程序的,是具有数个结点并且这些结点由互连网络结合在一起的并行处理机。A-NET 计算机正在研制中,该机的各个结点均由 PE 和 router 两部分组成^[1-6]。PE 用于执行加载到本结点内的对象中的 method, router 用于与其它结点间的通信^[5]。

为了对 A-NETL 所描述的程序进行逻辑跟踪,获得设计 A-NET 机所需要的硬件数据,我们利用 UNIX 的进程管理和进程通信功能开发了 A-NET 机的模拟程序。

在 A-NET 机中,host 与各结点间、各结点相互之

间、节点内 PE 与 router 之间均具有并行性,因此考虑让 A-NET 机中具有并行性的各个部分分别和具有该部分功能的进程相对应,利用 UNIX 的系统调用 fork(), exec() 来实现构成模拟程序的若干个进程的产生。

当对 n 个结点进行模拟时,router 进程被 host 进程启动后,参照选定的互连网络拓扑信息,生成 n 个 router 进程。各 router 进程分别具有各结点的 router 的功能。生成后的 router 进程完成自身的初始化之后,分别生成与其一一对应的 PE 进程。因此,host 与 router 进程各为 1 个,而 router 进程和 PE 进程各为 n 个。

host 进程与各 router 进程、各 router 进程和本结

^{*} 本文工作获日本学术振兴会资助。

理器性能时微处理器设计策略将采用动态转移预测技术。

结论 理想的转移预测机制应该:(1)性能较高。因为在现代高性能微处理器中,即使 5% 的误预测率也会带来较大的性能损失,(2)不应因程序的不同表现出太大的差异。(3)占用硬件资源尽可能小。

静态转移预测性能较动态转移预测低,且不同的程序,预测准确率差异较大;动态转移预测要获得较高的性能,硬件开销大。所以单纯的转移预测方法不能很好地提高微处理器的整体性能。如果能够尽量减少微处理器的转移执行,多条件码和条件执行技术可以消除一些条件转移指令^[2]。对于必须执行的转移操作,再辅以转移预测机制,能够提高转移预测的准确率。因此

理想的方法是在硬件上支持多条件码、条件执行技术,尽量减少条件转移行为,对条件转移提前形成条件码,并采用转移预测机制来提高微处理器的整体性能。

参考文献

- 1 郑纬民,汤志忠. 计算机系统结构. 北京:清华大学出版社,1998
- 2 胡良校,方滨兴,胡铭曾. 转移预测的局限性及可能的解决方案. 计算机工程,1998,9:45~48
- 3 郑飞. 新一代 RISC 微处理器的结构特征. 微处理机,1995,(4):1~6
- 4 郑飞. 超标量与超流水线混合结构微处理器 Pentium. 微处理机,1994,(4):1~5