

异构多核处理器分支预测器研究与设计

金玉苹¹, 焦 平², 佟 阳¹

(1 牡丹江师范学院 数学系, 黑龙江 牡丹江 157012;

2 哈尔滨工程大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

摘 要: 为解决单核处理器时钟频率难以提高、处理器功耗逐渐增加等问题, 文中提出了一种新型异构多核处理器的设计方案. 该结构中增加了 B-Cache 结构和 C-Core 控制器, 这种新型异构多核处理器避免了流水线因分支预测失误而 flush, 提高了整个处理器执行效率.

关键词: 异构多核处理器; B-Cache; 动态指令发射

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2011)11-0137-04

The Research and Design of Branch Prediction Based on Multicore Heterogeneous

JIN Yu-ping¹, JIAO Ping², TONG Yang¹

(1 College of Mathematics, Mudanjiang Normal University, Mudanjiang 157012, China;

2 College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: Aiming at the problem that it was difficult to improve the processor performance only by improve the frequency of the clock, as well as superscalar pipeline stall, this paper introduced an architecture of heterogeneous multi-core processor which used B-Cache structure and C-Core processor controller. The architecture avoided the pipeline flushed due to branch miss—predict. So the architecture could improve overall efficiency of Multi-Core processor.

Key words: CMP; B-Cache; dynamic instruction issue

1 引言

多核处理器技术的出现很好地解决了处理器功耗和性能的问题. 通过多核处理器的并行计算技术, 不仅可以大幅度提高处理器计算能力, 同时也可以降低处理器的功耗. 目前, 多核架构已经从通用的对称设计迁移到“主核心—协处理器”的异构架构. 通常异构多核处理器架构只有一个或多个通用核心用以承担任务的指派功能, 而诸如协处理器、加速器和外设等的功能都可以由专门的硬件核心来完成. 因此, 通过功能的细化和分工可以实现多核处理器执

行效率和性能的最大化. 现在, 较流行的多核处理器普遍采用同构对称结构^[1,2], 该结构中每个核心会有自己的一级 Cache, 而多个核心则通过公共总线共享二级 Cache.

对于同构多核处理器结构, 处理器设计的复杂度虽然可以降低, 但是在处理器内核工作频率不提高的条件下提升处理器的性能, 仅仅是增加同构处理器的内核, 存在着一定的极限. 当这个极限值达到后, 内核数量的增加已无法再进一步提升处理器的性能. 这就是著名的 Amdahl 定律. 虽然同构处理器内核的不断增多能够加强并行处理能力, 但软件中

收稿日期: 2010-12-21; 修回日期: 2011-01-15

基金项目: 牡丹江师范学院科学技术研究项目(QY201001); 黑龙江省青年科学基金项目(QC2010044); 牡丹江师范学院大学生创新性实践重点项目; 牡丹江市科学技术计划项目(G2010q810)

必须串行执行的指令数目必然会制约着整个系统的处理性能。

因此,为了提高处理器的性能,突破同构多核处理器的限制,文中设计了一种新型异构多核处理器结构,该结构中增加了预测失误恢复器 B-Cache (Branch instruction Cache, B-Cache). 这种异构多核处理器不仅可以优化处理程序中大量的 loop 程序,同时采用了不间断流水线技术从而避免了流水线因分支预测失误而导致 flush. 因此,文中新型异构多核处理器结构在充分享用多核处理器资源的同时,提高了整个处理器执行的效率。

2 多核处理器现状

目前决定多核处理器性能好坏的关键技术是核间结构的设计与采用何种核间通信机制. 实践证明只有在适当的多核处理器结构下,同时采用与结构相适合的通信机制,才能真正地发挥多核处理器性能。

另外,当采用 GAs 两级动态分支预测器后,预测精度已经很高,分支预测的准确率可以达到 80%~95%. 但是,每次分支预测失误会带来至少 3 个时钟周期的消耗. 预测失误带来的损失也将随着分支指令数的增加而逐渐增加。

2.1 多核结构研究

多核处理器结构主要分同构多核处理器结构和异构多核处理器结构两类,同构是指处理器内部的每个核的结构都是完全相同的,异构是指核的内部结构是不完全相同的。

图 1 是 IBM Cell 异构多核处理器结构图^[3]. Cell 具有一个运行 Power 指令的主核 PPE 和 8 个 SIMD 辅助核 SPE,通过一条高速总线 EIB 进行连接,其中 PPE 包括一个 64 位、双发射、双线程、顺序执行的运算核心,可以同时提取 4 条指令和两个结果,负责运转操作系统;而 SPE 则负责完成具体的浮点运算、矩阵运算、科学计算、多媒体处理等数据处理能力。

这种异构多核处理器在 3.2 GHz 的频率下单精度浮点运算的峰值可以达到 201 GFLOPS,而同频率的 Intel Pentium4 处理器的峰值仅 25.6 GFLOPS,这说明了异构多核处理器比同构多核处理器具有一些先天上的优越性^[4]. 而这正是文中采用异构多核处理器的原因之一。

2.2 数据重用

数据重用技术起源于记忆法是一种优化程序技

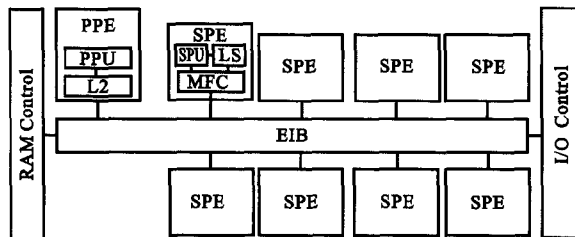


图 1 Cell 结构图

术. 记忆法可以自动追踪记录程序在执行过程中各种复杂的计算结果,并且在条件允许的情况下,记忆法可以重用记录过的结果来避免大量复杂的计算。

图 2 是数据重用机制工作原理图,取指部件每取得一条指令,就会启动数据重用机制,重用机制先判断所取指令是否可重用候选,如果是重用候选则继续判断是否为重用测试并且和历史实例相匹配,则数据重用机制重用以前结果并将已经取来的指令丢弃,再进行新的取指. 否则正常执行指令并将其结果记录为历史实例输出。

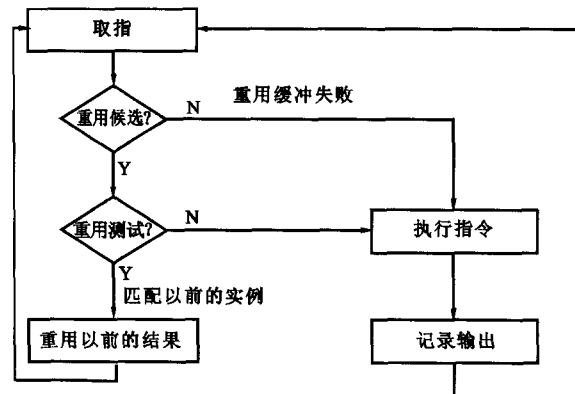


图 2 数据重用机制原理

3 一种异构多核结构

针对现有异构多核处理器结构中分支预测失误恢复问题,本文在对数据重用技术的研究分析基础上,设计并改进了一种新型异构多核处理器结构,如图 3 所示. 新型结构采用了 5 个处理器内核, C-Core 核负责指令的调度,另外 4 个 E-Core 辅助核负责指令的执行。

指令执行过程如下:

(1) 分支预测器负责 i-cache 的取指操作. 当取分支指令,将分支预测器预测的相反路径上取指,并将取出后续指令序列送入到 b-cache 中,同时对已经取到的指令进行译码操作(伪译码操作)。

(2) Issue Station 发射站负责将 i-cache 中的指令发射到 C-Core 处理器中进行指令的调度。

(3) C-Core 功能是将 Issue Station 发射站发

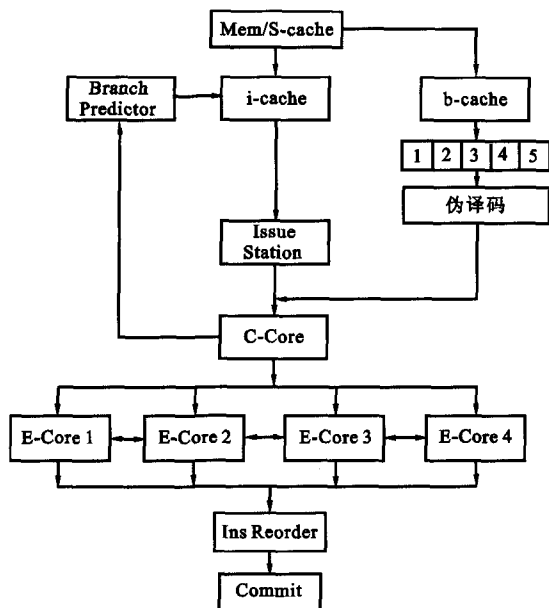


图3 异构多核处理器结构

射来的指令进行标记分析后,根据C-Core调度算法同时依据各个E-Core中指令的状态信息,将指令分配到相应E-Core中等待执行。

(4) 4个E-Core负责执行被调度来的指令序列。

(5) 指令站对将流入到此站中的乱序指令序列,按照指令流的初始指令序列进行排序,保证指令在下一个处理器时钟周期内由Commit对其按序提交。

(6) Commit指令负责提交经Ins Reorder指令站排序的指令序列。

4 分支预测失误恢复器

目前,由于采用各种技术使得分支预测的准确率可以提高到83.6%~97.8%^[5]。随之而来的是每次分支预测失误至少会带来3个时钟周期或者更多的消耗。正是由于分支预测失误恢复的问题,使得处理器的性能相应下降。所以为了解决这个难题,文中提出并设计了一种B-Cache预测失误恢复方案,并在此基础上利用数据重用技术,新的方案可以减少分支预测失误所导致的时钟周期消耗。

Branch Predictor“舍弃路径”上的指令序列被B-Cache送到B-Cache指令缓存中,并且对分支指令进行标记,例如tag1(标记舍弃路径上的指令序列,tag1为分支指令标记),n(n为超标量流水线的宽度)被定义为取指规模。再通过伪译码器对B-Cache获得的指令进行伪译码操作,经过伪译码的伪指令序列暂存到伪译码指令缓存中,然后根据C

-Core调度算法将确定执行的分支指令送到E-Core中,如果E-Core产生分支指令执行失效信号,那么B-Cache分支预测失误恢复器将在伪译码缓存的伪指令中查找相应的tag标记,找到后从伪译码缓存中将该已译码的指令队列移入到相应的E-Core。这种新机制与以往的流水线停顿、“重新取指”的恢复机制相比,至少可以节省2个时钟周期的消耗。

5 C-Core 控制处理器

C-Core控制处理器是整个异构多核分支处理器结构中最关键的一部分。C-Core不仅仅因为能够接收从Issue station发射过来的指令,同时还对指令进行二次处理,而且又依据C-Core的调度策略并对E-Core进行指令调度。正是因为C-Core规范的调度策略使得E-Core间才能高效、正确的并行处理工作。

5.1 动态指令发射技术

C-Core控制处理器根据动态指令发射技术^[6],再依据各个E-Core中指令执行的状态,动态的处理控制E-Core中的指令序列。当分支指令被C-Core获得后,配合执行分支指令流,C-Core能够计算出各个处理器的负荷状态,并从中选择一个负荷较小的处理器去单独执行分支预测在相反路径上的指令序列,同时将此E-Core进行标记处理。

E-Core控制处理器执行被调度来的分支指令时,当指令预测失误时,在下一个指令周期到来时E-Core中的指令流会被flush掉,但由于伪译码器中已经存在该分支指令的相反路径上的指令序列,并且已经完成译码工作。因此流水线不会因分支指令预测失误而导致停顿,分支失误带来的时钟周期惩罚也会减少。尽管如此,处理器的性能还是会受到分支预测失误的影响,探索一种更适合多核处理器的分支预测机制的改进策略成为必须。

5.2 C-Core 调度策略

C-Core调度算法不仅能够协调多个E-Core中指令流的执行,最重要的优势是能对已经执行的指令流序列进行多次优化和重组。调度算法^[7]主要考虑以下几个因子:E-Core间指令相关度R、E-Core指令执行饱和度S、E-Core间指令切换频率F、指令数N,分支指令预测正确率T以及分支指令预测失效率M。C-Core控制处理器调度指令时,主要依据下面两个公式进行计算。

E-Core 指令数的计算公式:

$$N = \frac{1}{R \times S \times F} \quad (1)$$

其中, R 表示 E-Core 处理器间指令的相关度, 相关度看作 1 指的是如果 E-Core1 中的一条指令和 E-Core2 中一条指令相关, 具体说如果 E-Core2 依赖 E-Core1 中的数据, 则 E-Core1 的入度为 1. R 表示每个 E-Core 中指令的入度占总相关度(总相关度为所有 E-Core 中入度和出度和的一半)的百分比.

S 表示 E-Core 指令执行饱和度, 即在核中正在执行的指令数占有所有 E-Core 中指令数总和的百分比, 如果将 E-Core 看成一条单流水线, 那么 S 反映的是这条单流水线的流动速度.

F 表示分支预测失误出现时, 在 E-Core 间切换数据频率的次数以此来提高处理器的性能.

E-Core 性能计算公式:

$$P = N \times \frac{T}{M} \quad (2)$$

其中, P 代表性能, T 和 M 分别表示分支指令预测的正确率与失误率. 在考虑分支预测失效的情况下, P 的计算结果反映出一个 E-Core 能达到的性能指标.

5.3 C-Core 算法的执行过程

图 4 表示 t 时刻每个 E-Core 中的指令所占百分比. 图 5 表示 t 时刻 E-Core 间指令相关度的统计. 由图 4 和图 5 可知, 利用式(1)和式(2)的计算结果, C-Core 能够均衡各个 E-Core 中处理指令的数目. 相应动态的分派各个 E-Core 核的指令数, 因而避免出现忙闲不均的低效率工作模式, 同时也能够使得总体任务的处理时间减少, 提高了 E-Core 控制处理器的处理效率.

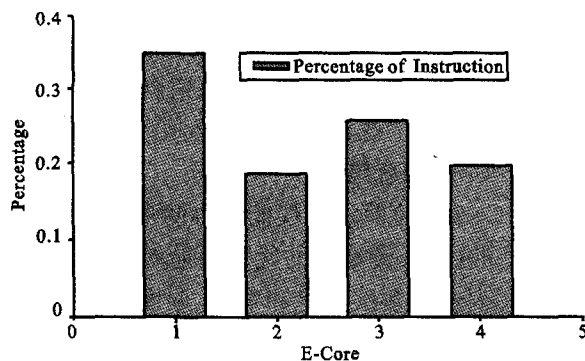


图 4 t 时刻每个 E-Core 中的指令数的状态

5.4 分支指令的执行

每条从 Issue station 发射来的指令经过 C-

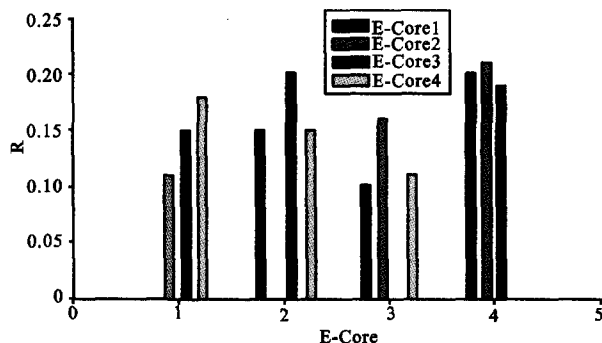


图 5 t 时刻 E-Core 间指令相关度

Core 检查后, 如果是分支指令, C-Core 会选择一个合适的 E-Core 去执行该分支指令的另一个路径上的指令序列. 式(1)和式(2)计算的结果成为选择 E-Core 的标准.

在一个已经标记过的 E-Core 中会执行分支指令的另一个路径上的指令序列, 标记的 E-Core 指令执行的结果最后能否提交取决于其它 E-Core 中的该分支指令的执行结果, 如果分支指令预测失误, E-Core 中执行的指令序列可以提交, 否则, 该标记过的 E-Core 中的指令序列会被 flush 掉.

6 结束语

文中设计了一种异构多核处理器结构. 由于采用一种新的异构结构而避免了因分支预测失误而产生的流水线 flush, 从而减少流水停顿等待的时钟周期, 提高了流水线效率. 而且, 为了平衡各个处理器负载平衡, 提出了 C-Core 指令调度算法, 经验证, 该算法不仅可以充分发挥出多核处理器的硬件优势, 还能充分利用多核处理器中的资源, 已成为提高处理器总体性能的主要途径.

参考文献:

- [1] McNairy C, Bhatia R. Montecito: A dual-core, dual-thread Itanium processor [J]. IEEE Micro, 2005, 25(2): 10-20.
- [2] 潘伟, 刘欢, 李广军. 一种新型高速低成本可重构 FFT 处理器结构[J]. 微电子学与计算机, 2008, 25(11): 29-32.
- [3] 多核系列教材编写组. 多核程序设计[M]. 北京: 清华大学出版社, 2007: 124-305.
- [4] Kahle J A. Introduction to the cell multi processor[J]. IBM Journal Res & Dev, 2005, 49(4/5): 589-604.
- [5] Tendler J M. POWER4 system microarchitecture[J]. IBM Journal of Research and Development, 2002, 46(5): 5-25.

(下转第 147 页)

```
memcpy(ldp_message_hello, plainBlock[8],8);
pal_mem_set (hello, 0, sizeof (struct ldp_message_hello));
/* 对消息头部分解码 */
ret = ldp_message_header_decode (pnt, size, &hello->
mes_h);
ret = ldp_tlv_common_hello_decode (pnt, size, &tlv_h,
&hello->common);
if (ret < 0)
return ret;
}
```

4 仿真实验

随机选择 50 个网络拓扑结构, 分别在未采安全策略及采用安全策略条件下用 NS-2 对安全性能进行仿真实验, 并进行归一化处理, 如图 10 所示. 分析可以看出, 采用安全策略后, MPLS 网络的安全性能提高了 11%~19% 之间.

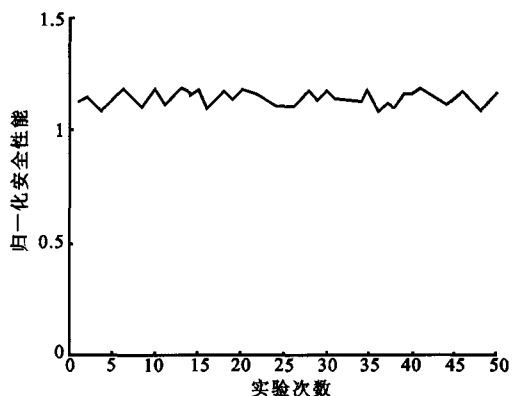


图 10 实施安全策略前后安全性能比较

5 结束语

文中指出了 BGP MPLS VPN 潜在的核心攻击, 给出了 LDP 加密/脱密模型. 采用 DES 算法对路由器之间会话所用的消息数据进行加密/解密, 只有被认证的路由器才能加入 BGP/MPLS VPN, 使得它的安全性能得到了提高.

参考文献:

- [1] IETF RFC—3031. Multiprotocol label switching architecture[S]. 2001.
- [2] RFC 3036. LDP Specification[S]. 2001.
- [3] IETF RFC — 3032. MPLS LabelStack Encoding [S]. 2001.
- [4] 阮立志, 许凌云. 一种基于网格计算环境的安全实现研究[J]. 微电子学与计算机, 2009; 265—268.
- [5] Schneier. 应用密码学: 协议、算法、与 C 源程序[M]. 北京: 机械工业出版社, 2000.
- [6] 吴伟. 下一代 IP 网络技术保障—多协议标签交换技术 [M]. 北京: 清华大学出版社, 2002.
- [7] 归奕红, 杜瑞颖. 基于 FGC 的无线传感器网络安全结构研究[J]. 微电子学与计算机, 2009(06): 223—226.

作者简介:

李海华 男, (1965—), 副教授. 研究方向为网络协议与信息安全.

(上接 140 页)

- [6] Kalla R. IBM Power5 Chip: A dual—core multithreaded processor[J]. IEEE Micro, 2004, 24(2): 40—47.
- [7] 奚杰, 陈杰, 朱玥. 利用 SystemC 实现多核系统的快速建模[J]. 微电子学与计算机, 2010, 27(7): 214—217.

作者简介:

金玉苹 女, (1981—), 硕士研究生. 研究方向为嵌入式系统结构设计、模式识别.

焦平 男, (1982—), 硕士研究生. 研究方向为嵌入式系统结构设计.