



(12) 发明专利

(10) 授权公告号 CN 102012841 B

(45) 授权公告日 2013.01.09

(21) 申请号 201010291185.8

(22) 申请日 2010.09.21

(73) 专利权人 上海大学

地址 200444 上海市宝山区上大路 99 号

(72) 发明人 赵文江 杨洪斌 吴悦 徐振坤

刘清华

(74) 专利代理机构 上海上大专利事务所(普通
合伙) 31205

代理人 何文欣

(51) Int. Cl.

G06F 9/48 (2006.01)

(56) 对比文件

CN 101526894 A, 2009.09.09,

US 20030014471 A1, 2003.01.16,

审查员 王思文

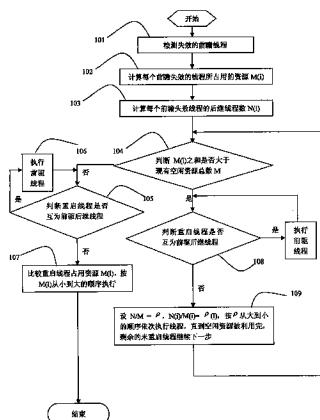
权利要求书 1 页 说明书 3 页 附图 1 页

(54) 发明名称

一种前瞻失效线程重启调度方法

(57) 摘要

本发明要解决的技术问题是提供一种前瞻失效线程重启调度方法。本方法是根据前瞻线程所占用的资源和相互之间的关系，决定前瞻失效线程重启的执行顺序。通过检测失效的前瞻线程；计算失效线程的资源和后继线程；判断他们之间的优先执行级别；再对其进行线程重启。本发明即能够满足优先重启资源需求少的线程，又能够考虑到该线程是其他线程的前驱线程和后继线程的情况，降低前瞻失效时的线程重启代价，提高了线程重启的效率。



1. 一种前瞻失效线程重启调度方法,其特征在于具体操作步骤如下:

A、检测失效的前瞻线程;

B、计算每个前瞻失效的线程所占用的资源,记为 $M(i)$, $i \in [0, n]$, n 为所有失效线程总数;

C、计算每个前瞻失效线程的后继线程数,记为 $N(i)$, $i \in [0, n]$, n 为所有失效线程总数;

D、判断步骤 B 中的 $M(i)$ 之和与现有的空闲资源总数 M 的大小,有选择的执行线程,具体步骤如下:

D1、当步骤 B 中的 $M(i)$ 之和小于现有的空闲资源总数 M ,就转到步骤 D11;当步骤 B 中的 $M(i)$ 之和大于现有的空闲资源总数 M ,就转到步骤 D21;

D11、判断重启线程是否互为前驱后继线程,如果是,先执行前驱线程,转到步骤 D11;否则转到步骤 D12;

D12、比较重启线程占用资源 $M(i)$,优先执行 $M(i)$ 小的线程;转到步骤 F;

D21、判断重启线程是否互为前驱后继线程,如果是,先执行前驱线程,转到步骤 D21;否则转到步骤 D22;

D22、对步骤 B、C 中的参数进行比较:当 $M(i) > M(j)$, $i \in [0, n]$, n 为所有失效线程总数,且 $N(i) < N(j)$ 时,就优先执行 $M(j)$ 线程;当 $M(i) < M(j)$ 且 $N(i) > N(j)$ 时,就优先执行 $M(i)$ 线程;当 $M(i) > M(j)$ 且 $N(i) > N(j)$,或者 $M(i) < M(j)$ 且 $N(i) < N(j)$ 时,设 $N/M = \rho$,把 ρ 称作“执行因子”, $N(i)/M(i) = \rho(i)$, ρ 越大,说明执行的优先级别越高,按 ρ 从大到小的顺序依次执行线程,直到空闲资源 M 被利用完,转到步骤 E;

E、对剩余的线程继续返回到步骤 D,直到所有失效的线程全部重新执行;

F、线程调度执行结束。

一种前瞻失效线程重启调度方法

技术领域

[0001] 本发明涉及计算机领域,提出了一种前瞻失效线程重启调度方法。

背景技术

[0002] 目前,高性能处理器仅仅提高处理器主频和复杂的指令流水线技术已经难以显著提高其性能,而且这样做还会导致高功耗和散热等棘手问题,因此,单片多核处理器(Chip Multi-Processors,CMP)技术成为提高处理器性能的关键技术之一。虽然CMP技术对并行执行多线程有很好的效果,但现在大多数程序还都是串行程序,如何将这些串行程序应用于CMP平台,并让其高效的运行,这些问题一直阻碍着高性能计算的发展。

[0003] 基于线程级前瞻(Thread-Level Speculative, TLS)的并行技术是解决这些问题的主要途径之一。但是前瞻技术推断得出的结果并不一定总是正确的,这表示有一部分线程必然会前瞻失效。当出现前瞻失效时,失效线程往往都是在同一时刻请求重启,但是由于处理器个数、内存块、寄存器等多个方面的影响,需要一个好的线程调度方法才能更好的提高线程重启的效率,因此前瞻线程重启调度方法成为串行程序并行化的关键技术之一。

发明内容

[0004] 本发明的目的在于针对已有技术存在的缺陷,提供一种前瞻失效线程重启调度方法,提高线程重启的效率。

[0005] 为达到上述目的,本发明的构思是:根据前瞻线程所占有的资源和相互间的关系,决定前瞻失效线程重启的执行顺序。通过检测失效的前瞻线程,计算失效线程的资源和后继线程,判断他们之间的优先执行级别,在对其进行线程重启。

[0006] 根据上述发明构思,本发明采用下述技术方案:

[0007] 一种前瞻失效线程重启调度方法,其特征在于具体操作步骤如下:

[0008] A、检测失效的前瞻线程;

[0009] B、计算每个前瞻失效的线程所占用的资源,记为 $M(i)$, $i \in [0, n]$, n 为所有失效线程总数;

[0010] C、计算每个前瞻失效线程的后继线程数,记为 $N(i)$, $i \in [0, n]$, n 为所有失效线程总数;

[0011] D、判断步骤 B 中的 $M(i)$ 之和与现有的空闲资源总数 M 的大小,有选择的执行线程;

[0012] E、对剩余的线程继续返回到步骤 D,直到所有失效的线程全部重新执行;

[0013] F、线程调度执行结束。

[0014] 上述步骤 D 线程的执行顺序直接依赖于步骤 B、C 中的参数,操作步骤如下:

[0015] D1、当步骤 B 中的 $M(i)$ 之和小于现有的空闲资源总数 M ,就转到步骤 D11;当步骤 B 中的 $M(i)$ 之和大于现有的空闲资源总数 M ,就转到步骤 D21;

[0016] D11、判断重启线程是否互为前驱后继线程,如果是,先执行前驱线程,转到步骤

D11 ;否则转到步骤 D12 ；

[0017] D12、比较重启线程占用资源 $M(i)$ ，优先执行 $M(i)$ 小的线程；转到步骤 F ；

[0018] D21、判断重启线程是否互为前驱后继线程，如果是，先执行前驱线程，转到步骤 D21 ；否则转到步骤 D22 ；

[0019] D22、对步骤 B、C 中的参数进行比较：当 $M(i) > M(j)$ ， $j \in [0, n]$ ， n 为所有失效线程总数，且 $N(i) < N(j)$ 时，就优先执行 $M(j)$ 线程；当 $M(i) < M(j)$ 且 $N(i) > N(j)$ 时，就优先执行 $M(i)$ 线程；当 $M(i) > M(j)$ 且 $N(i) > N(j)$ ，或者 $M(i) < M(j)$ 且 $N(i) < N(j)$ 时，设 $N/M = \rho$ ，把 ρ 称作“执行因子”， $N(i)/M(i) = \rho(i)$ ， ρ 越大，说明执行的优先级别越高。按 ρ 从大到小的顺序依次执行线程，直到空闲资源 M 被利用完，转到步骤 E ；

[0020] 本发明与已有技术相比较，具有如下显而易见的突出实质性特点和显著优点：本发明即能够满足优先重启资源需求少的线程，又能够考虑到该线程是其他线程的前驱线程和后继线程的情况，降低前瞻失效时的线程重启代价，提高线程重启的效率。

附图说明

[0021] 图 1 是本发明中的前瞻失效线程重启调度方法流程图。

具体实施方式

[0022] 本发明的优选实施例结合附图详述如下：

[0023] 实施例一：本前瞻失效线程重启调度方法，具体操作步骤如下：

[0024] A、检测失效的前瞻线程；

[0025] B、计算每个前瞻失效的线程所占用的资源，记为 $M(i)$ ， $i \in [0, n]$ ， n 为所有失效线程总数；

[0026] C、计算每个前瞻失效线程的后继线程数，记为 $N(i)$ ， $i \in [0, n]$ ， n 为所有失效线程总数；

[0027] D、判断步骤 B 中的 $M(i)$ 之和与现有的空闲资源总数 M 的大小，有选择的执行线程；

[0028] E、对剩余的线程继续返回到步骤 D，直到所有失效的线程全部重新执行；

[0029] F、线程调度执行结束。

[0030] 实施例二：本实施例与实施例一基本相同，特别之处如下：

[0031] 所述步骤 D 线程的执行顺序直接依赖于步骤 B、C 中的参数，操作步骤如下：

[0032] D1、当步骤 B 中的 $M(i)$ 之和小于现有的空闲资源总数 M ，就转到步骤 D11 ；当步骤 B 中的 $M(i)$ 之和大于现有的空闲资源总数 M ，就转到步骤 D21 ；

[0033] D11、判断重启线程是否互为前驱后继线程，如果是，先执行前驱线程，转到步骤 D11 ；否则转到步骤 D12 ；

[0034] D12、比较重启线程占用资源 $M(i)$ ，优先执行 $M(i)$ 小的线程；转到步骤 F ；

[0035] D21、判断重启线程是否互为前驱后继线程，如果是，先执行前驱线程，转到步骤 D21 ；否则转到步骤 D22 ；

[0036] D22、对步骤 B、C 中的参数进行比较：当 $M(i) > M(j)$ ， $j \in [0, n]$ ， n 为所有失效线程总数，且 $N(i) < N(j)$ 时，就优先执行 $M(j)$ 线程；当 $M(i) < M(j)$ 且 $N(i) > N(j)$ 时，就

优先执行 $M(i)$ 线程 ; 当 $M(i) > M(j)$ 且 $N(i) > N(j)$, 或者 $M(i) < M(j)$ 且 $N(i) < N(j)$ 时, 设 $N/M = \rho$, 把 ρ 称作 “ 执行因子 ”, $N(i)/M(i) = \rho(i)$, ρ 越大, 说明执行的优先级别越高。按 ρ 从大到小的顺序依次执行线程, 直到空闲资源 M 被利用完, 转到步骤 E;

[0037] 实施例三 : 本实施例所述方案的应用环境是面向多核处理器的前瞻线程并行执行环境, 本实施例不限定所述技术方案应用环境中多核处理器架构和线程调度方式。

[0038] 参照图 1, 本前瞻失效线程重启调度方法的操作步骤如下 :

[0039] 步骤 A(101)、检测失效的前瞻线程。

[0040] 步骤 B(102)、计算每个前瞻失效的线程所占用的资源 $M(i)$ 。

[0041] 步骤 C(103)、计算每个前瞻失效线程的后继线程数 $N(i)$ 。

[0042] 步骤 D1(104)、判断 $M(i)$ 之和是否大于现有空闲资源总数 M 。

[0043] 步骤 D11(105)、判断重启线程是否互为前驱后继线程。

[0044] 步骤 D11(106)、执行前驱线程。

[0045] 步骤 D12(107)、比较重启线程占用资源 $M(i)$, 按 $M(i)$ 从小到大的顺序执行。

[0046] 步骤 D21(108)、判断重启线程是否互为前驱后继线程。

[0047] 步骤 D22(109)、设 $N/M = \rho$, 把 ρ 称作 “ 执行因子 ”, $N(i)/M(i) = \rho(i)$, ρ 越大, 说明执行的优先级别越高, 按 ρ 从大到小的顺序依次执行线程, 直到空闲资源被利用完, 剩余的未重启线程继续执行下一步操作。

[0048] 以上对本一种前瞻失效线程重启调度方法, 进行了详细的介绍。本文结合说明书附图和具体实施例进行阐述只是用于帮助理解本发明的方法和核心思想 ; 同时, 对于本领域的一般技术人员, 依据本发明的方法和思想, 在具体实施方式和应用范围上均会有改变之处, 综上所述, 本说明书内容不应理解为对本发明的限制。

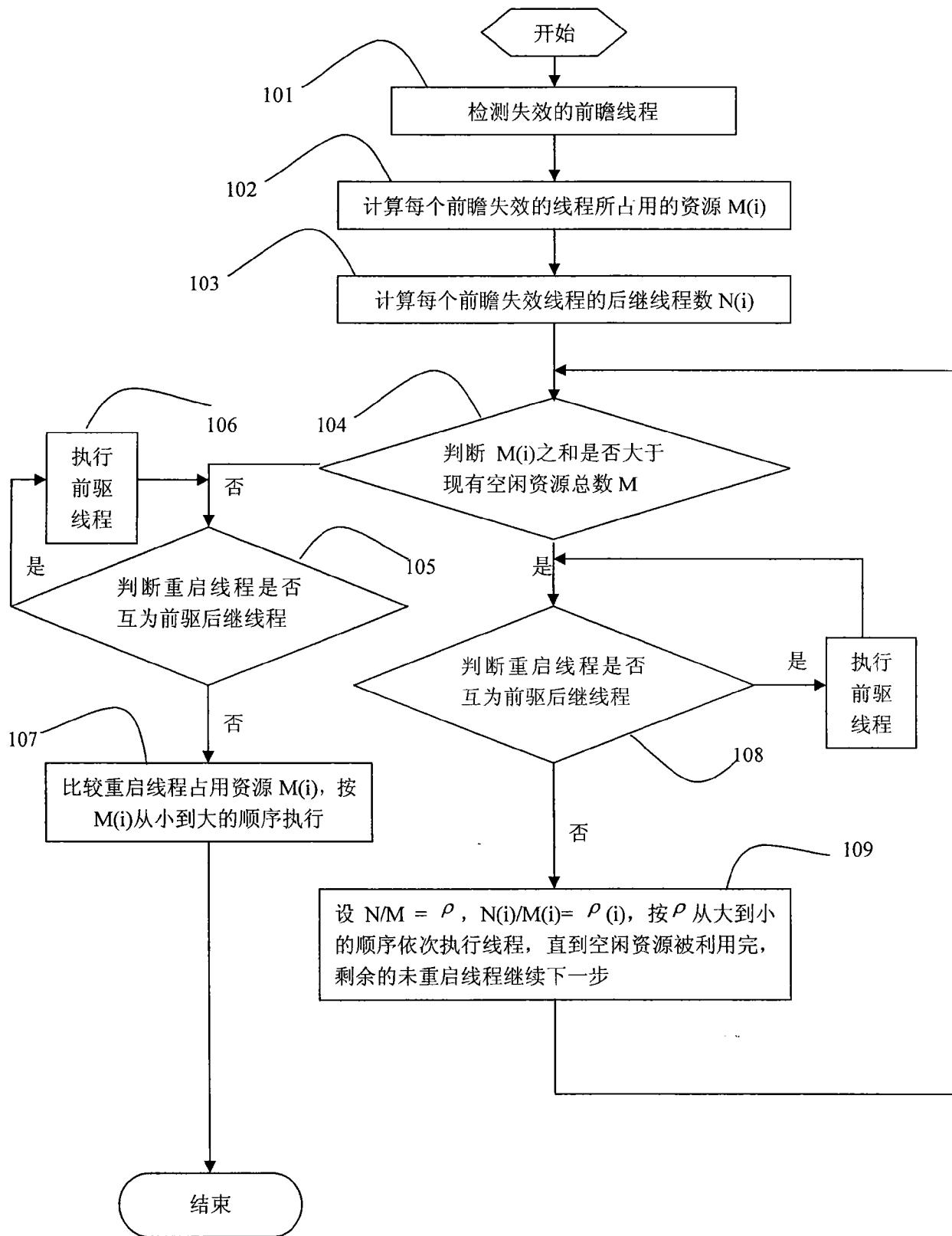


图 1