

地球引力位函数在流处理器上的实现与分析

曹 维, 车永刚, 王正华

(国防科技大学计算机学院并行与分布处理国家重点实验室, 长沙 410073)

摘要: 流处理器是新型高性能微处理器的代表之一。该文通过分析 Imagine 流处理器体系结构, 实现某卫星轨道预报系统核心子程序 CBPTRB, 对基于 Imagine 的钟精确模拟器 ISim 进行性能测试, 并将结果与该程序在普通 x86 处理器上的性能进行对比, 同时根据实验结果提出改进方法。

关键词: Imagine 流处理器; 流编程模型; CBPTRB 程序; 性能评估

Implementation and Analysis of Gravitational Potential Function on Stream Processors

CAO Wei, CHE Yong-gang, WANG Zheng-hua

(State Key Lab of Parallel and Distributed Processing, School of Computer, National University of Defense Technology, Changsha 410073)

【Abstract】 The stream processor is one of the representative for high performance processors. On the basis of analyzing the structure of Imagine stream processor, this paper implements the kernel procedure of a satellite orbit predicting system called CBPTRB. The performance of the cycle-accurate simulator based on ISim is tested, and the results of which is compared with the one getting from general x86 processor. According to the experimental results, the improved method is introduced.

【Key words】 Imagine stream processor; stream programming model; CBPTRB program; performance evaluation

1 概述

Imagine 是斯坦福大学设计的一款面向媒体处理的可编程流处理器, 用作通用处理器的协处理器。该处理器能在单精度应用上达到 16×10^9 float/s 的峰值性能, 而在 16 bit 应用上可达 32×10^9 float/s 的峰值。在一系列真实应用中, Imagine 能维持每周期 50 条指令及 15×10^9 float/s 的运算带宽。在设计中, 期望的 Imagine 运行频率为 400 MHz^[1]。该流体系结构设计初衷为加速流媒体处理过程中具有数据量大、运算密集、并行性较高的应用。研究表明, 一些科学计算, 如 FFT、Jacobi 迭代、Seidel 迭代能较好适应这种体系结构^[2-3]。CBPTRB 子程序由得克萨斯大学的 Bschutz 等人开发, 在某卫星轨道预报系统中, 该子程序用于计算地球非球形引力对卫星产生的加速度, 其运行时间占整个程序运行时间的 90% 以上, 对程序整体计算性能影响很大。鉴于该程序的重要性, 本文将其映射到 Imagine 流处理器上, 以试图加速其计算速度, 但测试结果表明, 由于存在较大的访存开销, CBPTRB 程序在 Imagine 上难以取得较好的性能。

2 Imagine 流编程模型及流处理器体系结构

2.1 流编程模型

媒体处理应用程序有大量的生产者-消费者局部性。流编程模型通过将应用表现为一系列用于产生和消费数据流的计算核心(kernel)以揭示这些特征, 每个数据流(stream)都是一组同一类型的数据记录, 计算核心对输入的流元素执行相同操作, 并生成一个或多个输出流。通过使用 2 种专门为 Imagine 开发的语言: StreamC 和 KernelC 进行媒体应用的编程。流程序通过 StreamC 流级程序指定核心的执行顺序, 并将数据组织成有序的流, 从一个计算核心传递至下一个。流程序中

计算核心使用的是 KernelC 核级程序, 它们被组织成循环的方式, 在每次循环迭代中处理来自输入流的元素, 并为输出流生成输出元素^[4-5]。

2.2 流体系结构

图 1 为 Imagine 流处理器体系结构。

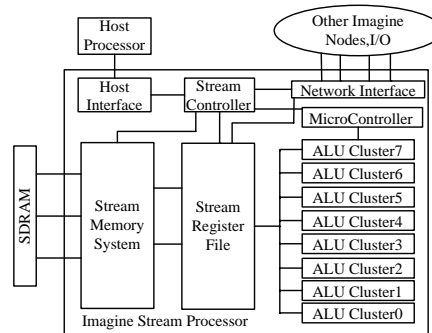


图 1 Imagine 流处理器结构

主处理器执行由 StreamC 流级程序编译得来的标量代码, 并通过片上的流控制器将流指令发送至协处理器 Imagine。流寄存器文件(SRF)是用于存储流的一块 32 KB 大小的存储区, 其分别与运算单元、片外存储器、网络接口和流控制器直接相连。所有操作数据的流指令传输都需通过

基金项目: 国家自然科学基金资助项目(60603055); 国家“863”计划基金资助项目(2007AA01Z116)

作者简介: 曹 维(1983 -), 男, 硕士研究生, 主研方向: 高性能计算机体系结构性能测评; 车永刚, 副研究员、博士; 王正华, 教授、博士、博士生导师

收稿日期: 2008-09-30 E-mail: caowei193@gmail.com

SRF 流的读取和存储指令使得流元素在 SRF 和内存中传输。KernelC 核级程序由一序列的 VLIW 指令组成, 并存放于微控制器上一个 2 000×576 bit 的 RAM 上。微控制器以 SIMD 的方式发送 kernel 指令至 8 个算术运算 cluster 上。每个 cluster 包括 6 个 ALU(加上一些有用的功能单元)以及分布在若干个本地寄存器文件(LRF)上的 304 个寄存器。cluster 中的每个功能单元的输入都是由单独的 LRF 提供。网络接口模块负责传送本节点的 SRF 和外部网络间的流^[1]。

3 CBPTRB 子程序特点及流编程实现

CBPTRB 子程序主要进行勒让德和缔合勒让德多项式计算。在某卫星轨道预报系统中, 该子程序用于计算地球非球形引力对卫星产生的加速度, 调用频繁, 运行时间占整个程序运行时间的 90% 以上, 对程序整体计算性能影响很大, 其代码主要由一些循环构成, 而且这类循环的形式都是对几个一维数组点积, 其中一些循环存在紧密的循环间相关, 虽然可以通过循环展开使依赖距离增大, 以实现 Cluster 间并行, 但是这样带来了冗余计算, 而一些循环如:

```
for( j=1;j<=nmtot ; j++){
    pvp = pvp + (tnm[j]*ppnm[j]);
    pvrp = pvrp + (tnm[j]*ppnm[j])*anp1[j];
}
```

不存在循环间相关, 除去存在循环间相关的一些循环后, 本文选取程序中的 5 个易于流化的循环 loop1, loop2, loop3, loop4, loop5, 当规模大小为 n 时, 各个循环所处理的向量长度分别是 $n, n, n(n-1)/2, n(n-1)/2, n(n-1)/2$, 在规模大于 40 的时候, 这些循环浮点运算次数之和占程序总数的一半以上。这些循环之间的输出和输入缺少相关性, 不容易组织成核级程序的串行调用。在其内部对向量各个元素的读入相互独立, 易将其组织成数据级并行。对这 5 个循环分别进行流化, 分别对应核级程序 kernel, kernel1, kernel2, kernel3, kernel4。以上述循环为例, 将向量 $tnm, ppnm, anp1$ 分别组织成 kernel3 的输入流 stnm, sppnm, sanp1, 每个运算 cluster 每次循环得到流的一个元素, 并对这些元素进行加乘运算, 直至对整个输入流循环处理完毕。这些核级程序, 除了 kernel 与 kernel1 之间, 由 kernel 产生的输出流作为 kernel1 的输入流外, 其他各个核级程序之间基本上没有生产者-消费者的关系, 每个核级程序处理的流都不完全相同, 每个数据流通过 1 个或 2 个核心完成计算过程。

4 实验

4.1 实验环境

采用 Stanford 大学针对 Imagine 处理器所编写的时钟精确模拟器 ISIM 对上述程序的性能进行模拟测试, 目标 Imagine 处理器采用典型配置, 主频 400 MHz。作为对比, 选择一个基于 Intel 32 bit 处理器的平台来测试对应的 C 程序性能, 该平台配置为: CPU 为 Intel Core Duo T2300 1.66 GHz; L1 Cache 为 2×32 KB instructions / 2×32 KB data; L2 Cache 为 2 048 KB; 主存为 PC5300 DDR2 1 GB; 编译器为 Intel C++ Compiler 9.0 for IA32 Optimizations: /O3 /QxB; 操作系统为 Windows XP Professional SP2。

4.2 实验结果分析

图 2 为各个循环在 Intel 处理器上运行周期数相对 Imagine ISim 模拟器上运行周期数的比值。由于两者的工作频率比为 400 MHz:1.66 GHz=1:4.4, 欲使加速比大于 1, 程序周期数必须控制在串行程序的 1/4。实际上, 从图 2 可见程

序在 Imagine 上运行周期数远大于串行程序运行周期数。串行程序在编译器的支持下, 可以将数组向量化, 并将单重循环进行循环展开或软流水, 因此可以获得较好的性能。而对流化程序, 从各个核级程序的输入输出流关系可以看到, 由于每个 kernel 程序所处理的流不尽相同, 使每次调用 kernel 程序都必须重新加载流, 而加载流的开销较大。各个流之间的生产者-消费者关系不明显, 每个流通过 1~2 个核心就完成了计算, 这使程序不能充分利用流体系结构的数据重用性。

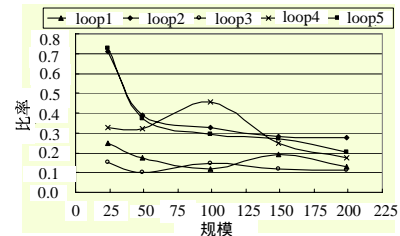


图 2 周期数对比

图 3 为不同规模下, 各个循环所达到的浮点运算性能。由此可见, 在规模较大时(流长度为 10^5 左右时), loop6 能达到约 490×10^6 float/s 的性能。与 Imagine 体系结构的峰值性能 7.96×10^9 float/s 相比, 仅约为峰值的 6.16%。FFT、Jacobi 迭代、Seidel 迭代等程序都是对多维数组进行操作, 拥有多重循环, 从流化方法能开发指令级并行和数据级并行, 并在此基础上充分利用 Imagine 体系结构。与这些程序相比, CBPTRB 程序处理的是些一维数组, 且没有嵌套循环, 无法较好地实现访存的延迟隐藏, 且在流化过程中, 只利用程序的数据级并行, 即将该级并行在 cluster 间实现, 流级程序分别调用核级程序进行处理, 这增加了程序间通信, 降低了系统有效利用率, 而当规模更大的时候(流长度超过 10^5 时), 由于 SRF 无法存储所有数据, 需要将数据存储在内存, 因此尽管存在双缓冲机制, 但从内存中读取数据使得运算 cluster 仍然出现大量的 stall, 导致性能降低。

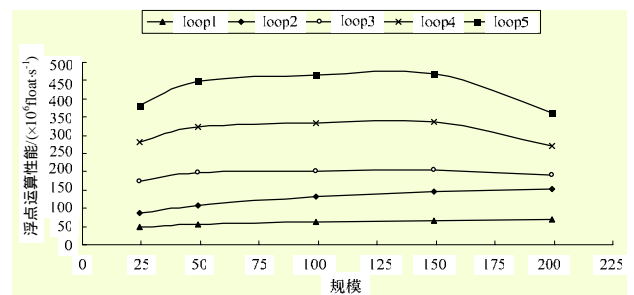


图 3 浮点运算性能

图 4 显示了不同规模下流级程序总体的准备开销、运行开销和 stall 开销, 图 5 显示了不同规模下各循环的开销。从图 4 和图 5 可以看出, 在不同流规模下, 各核程序真正用于执行计算的周期数都很少(不超过总周期数的 10%), stall 周期数也比较少, 相对较多的是为各个核程序准备流、并将流送至 SRF 中所需要的周期数(最坏情况下占总周期数的 90%)。

流处理器的浮点运算能力很强, 因此, 这几个 kernel 的浮点运算量相对来说较小, 但各个核级程序缺乏相互间的联系, 每次运行时都需要重新准备流, 这需要很大的时间开销, 特别是在图 4 中, 由于各个操作时间上存在重叠, 使得总的周期数比例超过了 1%。

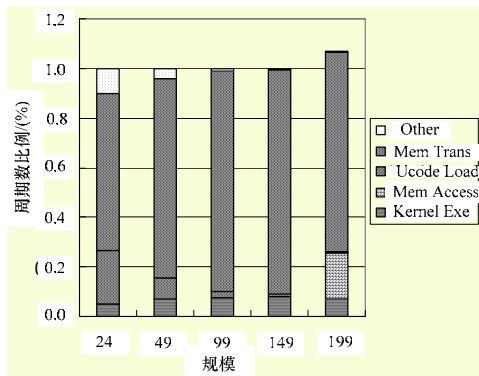


图4 程序开销分布

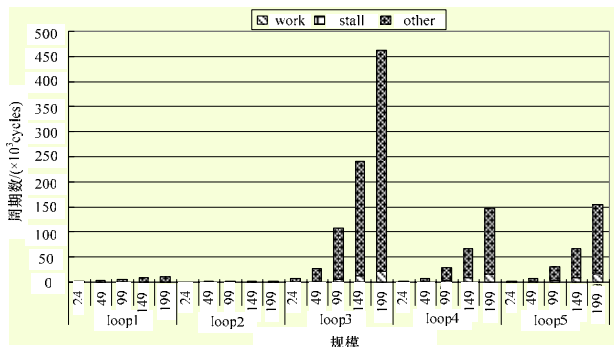


图5 运行周期比例

对于流程图，将不使用优化，与核程序中的主循环采用循环展开、软流水等优化方式对比，各个 kernel 的运行时间如图 6 所示。

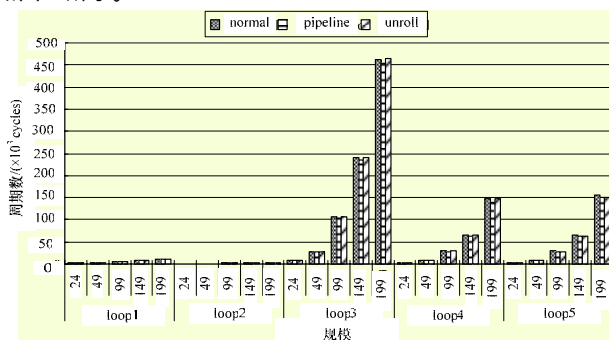


图6 优化周期对比

从图 6 中可以看出，进行 unroll 优化所得周期数比不进行优化的周期数略少，而进行 pipeline 优化所得的周期数要更少些，但总体来说，这 3 种方式所得的运行周期数几乎相同，核级程序真正用于计算的时间是很少的，而仅对这一部分的程序进行优化，所得到的效果必然不够明显。

5 结束语

流处理器是否适合科学计算类应用是研究者关心的热点。本文以地球引力位函数计算内核 CBPTRB 为例，对其进行流化并评测相关性能，包括执行周期数、浮点运算性能、程序开销分布、优化性能等。

实验结果表明，与 Intel Core Duo 处理器相比，Imagine 所需要的周期数更多，在 400 MHz 频率下所得到的加速比远低于 1，峰值性能也只有 490×10^6 float/s。由此可见，对于 CBPTRB 这类低计算访存比的程序，Imagine 难以获得好的性能。欲在 Imagine 取得较好的性能，不仅必须选取并行度高，计算/访存比高的程序进行流化处理，而且需要在映射方法上进行研究，将并行化的思想引入进来，充分利用流编程模型所提供的各种局部性，开发 Imagine 强大的计算能力。这对于以后算法的映射具有一定借鉴意义。

参考文献

- [1] Kapasi J K, William J, Dally S, et al. The Imagine Stream Processor[C]//Proc. of the 20th Int'l Conf. on Computer Design. Washington D. C., USA: IEEE Computer Society, 2002.
- [2] William J, Dally S, Khailany B, et al. Evaluating the Imagine Stream Architecture[C]//Proc. of the 31st Int'l Symp. on Computer Architecture. Washington D. C., USA: IEEE Computer Society, 2004.
- [3] 王桂彬, 唐沛蓉, 唐玉华. 线性方程组迭代法在流处理器上的映射与分析[J]. 计算机工程与科学, 2007, 29(7): 88-90, 95.
- [4] John D, Owens J, Scott R, et al. Media Processing Applications on the Imagine Stream Processor[C]//Proc. of the 20th Int'l Conf. on Computer Design. Washington D. C., USA: IEEE Computer Society, 2002.
- [5] Peter M, Kapasi J K, Owens J, et al. Programming System User's Guide[Z]. (2002-05-11). http://cva.stanford.edu/classes/ee482s/docs/ips_user.pdf.

编辑 陈文

(上接第 247 页)

(1)球面全景图的设计。用本文的方法可以实现柱状的全景效果，这可以满足大多数机动目标跟踪视景仿真的需要。但是，在特殊情况下(如目标过顶飞行)，也需要构建球面的全景图。对此，目前 VRML 中没有有效的解决方法。

(2)CCD 跟踪中自动变焦的处理。很多 CCD 的光学变焦倍数并不固定，随目标的远近能够自动作出调整，同时，视角也会动态地变化。如何在 VRML 中建立精确的 CCD 自动变焦模型也是一个值得研究的问题。

参考文献

- [1] Ames A L, Ndaeu D R, Moreland J L. VRML 2.0 Sourcebook[M].

[S. 1.]: John Wiley & Sons, Inc., 1997.

- [2] 刘怡, 张洪定, 崔欣, 等. 虚拟现实 VRML 程序设计[M]. 1 版. 天津: 南开大学出版社, 2007-01.
- [3] 李占利, 孙秀英. 一种实现 VRML 中坐标变换的方法[J]. 西安科技大学学报, 2006, 26(2): 240-244.
- [4] 张洁玉. 图像拼接及柱面全景图生成算法研究[D]. 大连: 大连理工大学, 2006-02.
- [5] 丁伟, 胡于进, 王学林, 等. 精确控制 VRML 对象连续运动的一种方法[J]. 华中科技大学学报: 自然科学版, 2003, 31(6): 29-31.

编辑 索书志