文章编号:1001 - 2486(2006)04 - 0043 - 06

二维拉格朗日和欧拉结合法在流处理器 MASA 上的实现与评测*

张春元,文 梅,伍 楠,荀长庆,吴 伟
(国防科技大学 计算机学院,湖南 长沙 410073)

摘 要:现代半导体工艺技术的发展使得在单芯片上放置数百个运算单元成为可能,但是全局片上片外带宽受限。通用处理器体系结构不能较好地适应变化,仍然依靠全局片上结构,少量的运算单元。而流体系结构拥有大量的运算单元、鲜明的存储层次,使得在有限的片外带宽下,用高的本地带宽来满足大量运算单元的需求。首先介绍了原型 MASA 流体系结构,然后给出了爆轰流体力学中的二维拉格朗日和欧拉结合法(Ygx2)在流体系结构上实现的实例研究,最后用时钟精确的模拟器来评测应用的运行性能,结果表明 Ygx2 应用在 500MHz 的 MASA 上运行结果与 1.6GHz 的 Iantium2 的比较快近 4 倍,证实了流体系结构在高性能计算领域的极大潜力。

关键词:流体系结构;Ygx2;高性能计算
中图分类号:TP303 文献标识码:A

Implementation and Evaluating of a 2D Lagrange-Euler Method on MASA Stream Processor

ZHANG Chun-yuan, WEN Mei, WU Nan, XUN Chang-qing, WU Wei
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Modern semiconductor technology allows us to place hundreds of functional units on a single chip which provides limited global on-chip and off-chip bandwidths. General purpose processor architectures have not adapted to this change in the capabilities and constraints of the underlying technology, still relying on global on-chip structures for operating a small number of functional units. Stream processors, on the other hand, have a large number of functional units, and utilize multiple register hierarchies with high local bandwidth to match the bandwidth demands of the functional units with the limited available off-chip bandwidth. This paper describes the microarchitecture of MASA and presents the implementation of the application in fluid dynamics. We developed cycle-accurate simulator to evaluate the performance. The results show that the application on 500MHz MASA outperforms a 1.6G Itanium2 by a factor of 4. This research confirms that stream architecture has the potential to deliver high performance.

Key words: stream architecture; Ygx2; high performance computing

传统的程序设计模型和体系结构模型已经很好地运行了半个多世纪,对于从事计算机研究的人员来说,可谓根深蒂固。但是这个模型导致处理器越来越复杂,从而引发可靠性、功耗、计算效率、成本等问题。流是基于‘生产者 – 消费者’的计算模型,也是随计算机应用领域的发展而逐渐明晰的计算模型。流体系结构[1]具有强局域性、并行性、解耦合访存操作和计算等特征,适应了现代半导体技术的发展,特别适合于计算密集型的并行应用,使编译优化后的代码高性能地运行。在长期跟踪研究经典流体系结构①及基础上提出了多维可扩展自适应流体系结构(MASA)原型流体系结构[2-3],其适用的应用领域包括媒体处理、信号处理、科学计算等密集计算领域。流体力学中的二维拉格朗日和欧拉结合法(简称 Ygx2)是北京应用物理与计算数学研究所为评价高性能计算机的性能而提炼的 IAPCM Benchmarks 应用之一。本文介绍了 MASA 流体系结构,讨论如何在 MASA 上开发 Ygx2 的流式算法,并评价其性能。分析表明在科学计算领域的应用中大量的访存操作和流体系结构丰富的计算带宽之间存在算法上的折中

* 收稿日期:2006 - 01 - 17
基金项目:国家自然科学基金资助项目(60573103);高性能计算创新团队基金资助项目(IRT0446)
作者简介:张春元(1964—),男,教授,博士。
①我们所指的经典流体系结构是 Imagine Merrimac[4-5]

和优化,本文提出的 Ygx2 的流式算法对于在其他科学计算领域中的计算密集型问题在流体系结构上的映射有极大的借鉴意义,对流体系结构研究做出有益探索。

1 MASA 流体系结构

MASA 体系结构如图 1 所示。MASA 体系结构包含两个异构的处理器核,一个嵌入式标量处理器核(简称标量核),一个 64 位流处理器核(简称流处理核),流处理核以协处理器的方式工作。MASA 拥有三级带宽存储层次:本地寄存器文件(LRF①),流寄存器文件(SRF),片外存储 memory。流处理核由 8 个 cluster 组成大规模计算单元阵列,总共包括 32 个对等的双精度浮点乘加单元,8 个除法单元和 8 个查找表,所有的功能单元完全流水。

MASA 采用两级编程模式:流级和核心级。流级访存并准备流,定义 kernel② 的执行顺序,核心级执行 Kernel,即对流的数据进行计算。流级程序在标量核上执行,Kernel 在流处理核上执行。和 Kernel 执行相关的数据流准备、启动执行等操作由流控制器控制执行。kernel 编译打包成一段 VLIW 序列,并按序广播到 8 个 cluster 上以 SIMD 方式执行。Kernel 的输入流、输出流以及 kernel 之间的中间数据流通过 SRF 流转,一个 kernel 内计算的数据则由 LRF 提供。

MASA 流体系结构的两大优势是:一、由于解耦合访存与计算,访存操作可以与 Kernel 并行;二、层次化的带宽,越靠近运算单元的带宽越高,加上应用的密集计算特征,使得大规模的运算阵列可以充分运转。

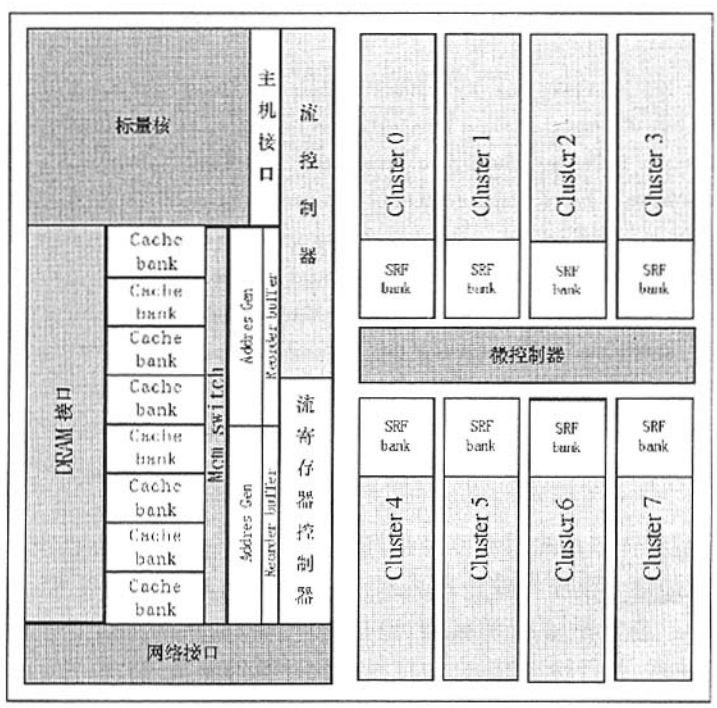

图 1 MASA 流体系结构

Fig.1 MASA stream architecture

2 YGX2 在流体系结构上的映射

求解二维爆轰流体力学问题没有一种国际公认的统一计算方法,只能针对不同的情况采用不同的方法。Lagrange 和 Euler 相结合的方法和纯粹的 Lagrange 方法相比,可以处理较大畸变的流体运动,而又比纯粹的 Euler 方法提供更精细的数值模拟结果。因此,常被用于处理介于拉格朗日和欧拉变形之间的

①分布在 cluster 中,图 1 中未画出。
②Kernel 指小程序,类似函数。

问题。该 Benchmark 基本计算过程使用差分方法求解偏微分方程,处理的网格大小为 64 × 603,需要迭代 537 次。流体系结构的编程模型为两级编程,流级和核心(kernel)级,分别用 StreamC 和 KernelC 语言编写,详见文献[12]。

2.1 实现难点

流式 Ygx2 算法的难点在于二维数据组的数据流组织和边界的处理。

Ygx2 中计算主要是根据网格中的邻近点来求解本点的值,例如通过某点邻近点的 x、r 来计算该点的 u、v 值。不同变量存在不同的对相邻点的需求情况,但相邻点最多是 8,也就是需要左右列和上下行的相邻数据。原来的 Fortran 程序利用二维数组处理,而流处理器中 8 个 cluster 一次读入一维流中连续的 8 个数据以 SIMD 方式执行,因此如何组织好数据并尽量减少数据冗余是高效运行的关键。考虑 SRF 大小为 512KB,数据流过长会导致 SRF 行为类似一个容量很小的缓存,不能发挥其捕捉生产者 – 消费者局域性的特点。因此将网格数据分块处理,每块规模为 24 × 150 个点。为了最大程度减少冗余数据,将每块数据按列组织,从左至右。每一列组织顺序如图 2 所示。8 个 cluster 一次读入流中连续的 8 个数据,因此对该数据流读 3 次,就可以将整列读入,并且相邻 3 行的点在一个 cluster 内,例如 cluster0 中存放 1、2、3,cluster1 中存放 4、5、6。在 cluster 中同时保存 3 列,每个 cluster 总存有中间行计算点所需的周围 8 个点,其他行的计算点所需要的点会有部分在其他 cluster 中,通讯是必要的。如此,就可以实现中间列的计算。接着读入下一列替换最左列即可实现窗口向右移动,计算过程类似。

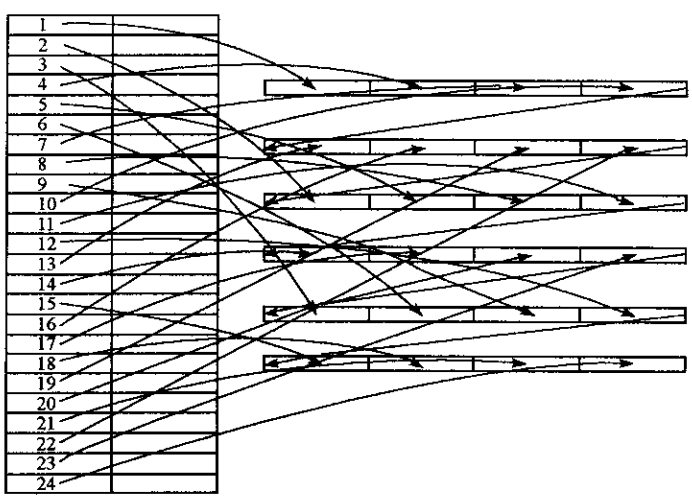

图 2 一块数据按列组织流的方式
(左图为正常列中点的排列,右图为流,顺序为由上至下,由左至右)

Fig.2 One column of data and it's stream format (the left figure is the mormal format, the right figure is the stream format)

边界问题主要有两种:(1)每个块的边界。对于每个子块第一行和最后一行由于无法获取相邻数据而计算不准确。采用冗余计算的办法。例如第一块计算 1 – 24 行,第二块计算 23 – 46 行,用第二块计算出来的 24 行结果替换上一块的输出。由于前面已经将流按列组织,一行的数据在流中不连续,而 kernel 里不能乱序处理流数据①,因此用标量程序来处理数据重组等过程。(2)整个网格的边界。整个网格的第一行和最后一行,程序一般会有特殊处理,而对于最左和最右列需要保持原来值不变。因此对于行边界,在块计算中标明第一块和最后一块,对第一块的第一行和最后一块的最后一行在 Kernel 计算时作特殊处理。最左和最右列则用标量程序对其进行重新赋值。

2.2 基本实现

因为 MASA 集成了一个传统标量核,流级程序运行在该标量核上,因此只将适合流处理的计算密集部分编为 kernel 程序交给流处理核来执行。基本上,按照功能划分 Kernel,开发了一个包含 18 个 kernel 的 Ygx2 流程序,假定初始和最终数据放在片外主存中。总的说来,因为在 kernel 里不能乱序处理流数

①文献[9]中提出索引流,在一定程度上解决乱序处理流数据的问题,但存在较大开销。

据,因此采用标量程序来处理数据重组等过程①。由于分块执行,因此每个 kernel 运行一次只处理一块数据,对每一块数据以及整个网格的处理基本流程如下:用标量数组为 kernel 运算准备输入数据,并赋值给流,并 load 到 SRF 中;运行 kernel,产生输出流;将输出流回存到数组中,由标量操作进行位置重组。Ygx2 中大部分的 kernel 获得一次整个网格点的相关变量需要运行 12 遍,得到整个网格的相关数据值后再进入到下一个 kernel。

在本程序中不同 kernel 之间的数据存在相关,而相同 kernel 之间的数据不存在相关性。而后者非常利于延迟隐藏,可以通过软件流水来优化。

2.3 局域性

MASA 流体系结构的带宽层次能捕捉两个层次的局域性:(1)LRF 级,Kernel 内部的局域性;(2)SRF 级,Kernel 之间的数据流的生产者-消费者局域性。与媒体计算不同的是,在 Ygx2 中体现的 kernel 之间的生产者-消费者局域性不明显,因为几乎每个 kernel 都有输出流需要回存到主存中重组,这一点将在第 4 节讨论。而在 Ygx2 应用中,每个 Kernel 内部操作产生的临时结果都存储在 LRF 中,只有输入流、输出流需要访问 SRF。因此对 Ygx2 应用,流体系结构所捕捉的主要是在 Kernel 内部的局域性。

3 性能评测

本文在 ISIM(Imagine 的时钟精确模拟器)的基础上实现了 MASA 的C++时钟精确模拟器以及 Xilinx Vertex4 FPGA 上的硬件仿真器[6-7],C++模拟器可以时钟精确地模拟整个 MASA 的运行行为,硬件仿真器主要是为了加速流处理器核中 ALU 阵列的模拟。模拟器参数为工作频率 500MHz,字长 64b,SRF 大小 512KB。表 1 是通过流式 Ygx2 程序模拟仿真所获得的部分实验结果。

表 1 Ygx2 应用部分实验结果

Tab.1 Some experiment data of stream Ygx2

	Memory 带宽 P (GB/s)	SRF 带宽 (GB/s)	LRF 带宽 (GB/s)	IPC②	ALU Gflops	浮点操作/访问 memory③
峰值	5.3	51.2	1248	40	36	
实际达到	2.4	7.3	384	24.2	18.4	76

表中可以看出应用的计算密集程度为 76 次浮点操作/访问 memory,数据访存依带宽层次分布,因此局域性较好,数据表明各级带宽设置可以满足要求。

丰富的数据并行、指令并行以及密集的计算操作使得 Ygx2 在 MASA 上获得相当高的性能,IPC 为 24.2,持续达到性能 18.4Gflops。应用执行时间为 11.8s。整个应用的执行时间由几个部分组成:kernel 在 cluster 内部的运行时间,包括 kernel 花在主循环的时间、非主循环时间和 cluster 停顿(等 SRF 传输数据);非 Kernel 运行时间,即 kernel 未被启动执行的其他应用执行时间,存在四种可能:加载 Kernel 微代码(ucode)、流控制器发射开销、标量核计算或传送流指令、等待主存 MEMORY 数据传输。图 3 表示整个应用的执行时间的分块图,由图可得 Ygx2 应用的约为 40% 的执行时间消耗在 Kernel 执行,Kernel 内的停顿极少。等待标量核的执行时间为 18.25%,等待访存时间为 37.23%,这

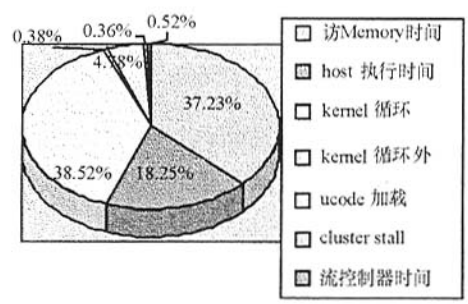

图 3 Ygx2 执行时间分布

Fig.3 Execution time breakdown of stream Ygx2

①实现数据重组的方法有几种,我们实现的这种方法在软件实现上较为容易,关于不同重组方法的效率评估不在本文讨论。
②包括非计算型指令。
③考虑了标量程序访存的因素。

是由于 Ygx2 中有些顺序访问和串行处理,无法隐藏延时。可以看出数据访问等开销大部分被隐藏,计算足够密集使得大规模的运算阵列能充分运行,较少出现等待数据的情况。

表 2 是 Ygx2 在不同机型上运行时间[8],比较结果可以发现 500MHz 流处理器的性能比 1.6GHz 的 Itanium(-O3-fast)快约 4 倍。

表 2 Ygx2 程序在不同机型上的运行时间

Tab.2 Performance of Ygx2 on other perocessors

	Alpha21264 500MHz	MIPS R14000 600MHz	Alpha21264 800MHz	Intel Itanium2 1GHz
时间	131.30 (s)	87.85 (s)	93.30 (s)	67.16 (s)
加速比	1.00	1.49	1.41	1.96
	Alpha21264 1.25GHz	Power4 1.3GHz	Intel Itanium2 1.6GHz	MASA 500MHz
时间	35.20 (s)	98.28 (s)	44.3 (s)	11.8 (s)
加速比	3.73	1.34	2.96	11.1

4 科学计算的新特征及未来的改进

经过研究流式 Ygx2,并与其他几种在 Merrimac 上实现的科学计算领域中的 StreamSPAS、streamFEM-3D、streamFLO、streamMD[9] 比较,发现这些科学计算呈现一些共同特征。例如 memory 访问与 SRF 访问、LRF 访问比率为 1:3:100 左右,而媒体处理该比率约为 1:10:200[10]。对片上流级存储 SRF 的访问,科学计算也显示相当的不规则性。对 SRF 的访问包括几类:(1)预取,片外 MEMORY 读入应用所需输入数据;(2)回存,存储应用的结果到片外 MEMORY;(3)流的重用,同样的流以同样的顺序被消费多次;(4)流非规则的重用,数据流从存储读入,重用多次,但是顺序变化;(5)流的生产者 – 消费者局域性,数据流被生产并以相同的顺序被消费;(6)非规则的生产者 – 消费者局域性,数据流被产生并多次消费,但消费顺序与产生的不同。图 4 显示在部分重要的密集型计算领域的几种典型应用中,各类访问的比率。

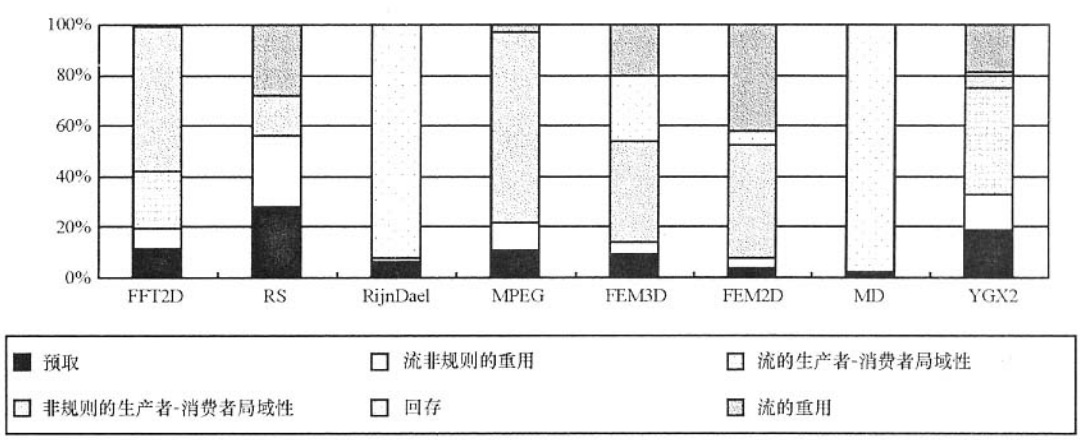

图 4 部分密集计算应用的流级存储访问分布

Fig.4 Stream level memory access breakdown of some intensive-computing applications

由图 4 可以看出,实现的流式 Ygx2 中存在较多的不规则生产者 – 消费者局域性,产生的数据并不以原来的顺序被使用,这是由于对 SRF 的访问必须是按流的顺序以及 Ygx2 的计算特征决定的。采用标量来处理数据重组,使得生产者 – 消费者局域性较少。

目前看来这些应用都在流处理器上获得了极好的性能[9,11],说明流处理所需要的应用主要特征是计算密集和数据并行,至于生产者 – 消费者局域性则没有强的要求,当然用分块来减少中间结果大小的方法是必要的。SRF 是限制单片 MASA 处理能力的主要因素,大规模的计算可以考虑在多片 MASA 上分任务执行。

结合科学计算的新特征,未来可以在几个方面进行调整。硬件方面,对SRF进行调整,如果SRF支持乱序访问,则Ygx2的生产者-消费者局域性则可以体现出来,可以进一步减少访存提高性能。软件方面,编译重负载的Kernel,程序员的向量化编程语言则是我们下一步的工作目标。

5 结论

流体系结构是一种利用大规模运算单元进行密集计算的结构,Ygx2程序具有丰富的数据并行和密集的计算,但映射在MASA这样的SIMD流处理器上存在科学计算所共有的多维数组计算带来的数据流难以组织,边界难以处理等问题。本文首先描述了MASA体系结构,然后详细介绍实现方法,包括流的组织,Kernel的窗口移动计算,分块,软流水等。然后基于实验数据对结果进行评测,数据表明在500MHz的MASA上运行该应用的时间比1.6G的Iantium2快约4倍。必须要说明的是,Ygx2在MASA上的加速来源于流体系结构的经典特征,而MASA很好地继承这些特性,因此能够很好地开发Ygx2的延迟隐藏、并行、局域性,最终获得很高的性能。本文为在流体系结构上映射一些其他的科学计算应用或媒体信号处理应用提供参考,为更好地开发面向科学计算的流体系结构积累第一手资料。

参考文献:

[1] Workshop on Streaming Systems[EB/OL]. [2003-08-22]. http://www.cag.lcs.mit.edu/wss03/.

[2] 文梅,伍楠,张春元,等. Multiple-dimension Scalable Adaptive Stream Architecture[A]. Ninth Asia-Pacific Computer Systems Architecture Conference,Beijing,2004,9:199-211.

[3] 文梅,伍楠,张春元,等. Multiple-morphs Adaptive Stream Architecture[J]. Journal of Computer Science and Technology,2005,20(5):635-646.

[4] Ahn J H, et al. Evaluating the Imagine Stream Architecture[J]. ACM SIGARCH Computer Architecture News,2004, 32(2):14-25.

[5] Erez M, et al. Analysis and Performance Results of a Molecular Modeling Application on Merrimac[A]. 2004 International Conference for High Performance Computing,Networking,Storage,and Analysis,USA,2004.

[6] 伍楠,文梅,张春元,等. A Stream Architecture Supporting Multiple Stream Execution Models[A]. Asia-Pacific Computer Systems Architecture Conference 2005,Singapore,2005,10:143-156.

[7] MASA工作组[EB/OL]. [2006-01-10]. http://masa.nudt.edu.cn.

[8] 袁国兴,等.评几种高档微处理器在运算科学计算问题时的性能[EB]. http://www.ccw.com.cn.

[9] Jayasena N S. Memory Hierarchy Design for System Computing[D]. Stanford PhD thesis,2005,9.

[10] Rixner S. Stream Processor Architecture[M]. Kluwer Academic Publishers, Boston,MA,2001,11.

[11] 文梅,张春元,伍楠,等. A Parallel Reed-solomon Decoder on the Imagine Stream Processor[A]. Second International Symposium on Parallel and Distributed Processing and Applications,Hong Kong,2004,12:28-33.

[12] 伍楠,文梅,张春元,等.面向Imagine流存储层次的程序设计模式研究[J].计算机研究与发展,2004,(12):328-333.