

Jiang Wenqian, Zhong Dingkun, Xie Zhangpeng, Wang Qun, Feng Xueshang. CPU/GPU-based parallel models of coronal polarization brightness calculation. *Chin. J. Space Sci.*, 2011, **31**(1): 51-56

基于 CPU/GPU 的日冕偏振亮度并行计算模型^{*}

江雯倩¹ 钟鼎坤² 解张鹏¹ 王群¹ 冯学尚²

1(中国地质大学信息工程学院 北京 100083)

2(中国科学院空间科学与应用研究中心 空间天气学国家重点实验室 北京 100190)

摘要 三维磁流体力学 (MHD) 数值模拟是用来研究日冕和太阳风最常用的方法之一, 其中将计算得到的日冕电子数密度转化为日冕偏振亮度 (Polarization Brightness, PB) 是与观测对比的重要方法。由于待转换电子数据网格密度、PB 数据网格密度和计算模型的复杂度, 使得日冕偏振亮度的计算比较耗时, 利用单 CPU 计算无法达到近实时转换日冕偏振亮度的要求, 从而影响了数值模拟的验证效率。本文在 CPU/GPU 环境下, 利用 CUDA 编程技术, 提出了一个日冕偏振亮度并行计算模型。实验结果表明, 该模型比 CPU 上的串行模型计算速度提高了 31.86 倍, 达到了近实时模拟与观测数据比对的计算要求。

关键词 日冕偏振亮度, GPU, CUDA

中图法分类号 P 353

CPU/GPU-based Parallel Models of Coronal Polarization Brightness Calculation

JIANG Wenqian¹ ZHONG Dingkun² XIE Zhangpeng¹
WANG Qun¹ FENG Xueshang²

1(*Information Engineering Institute, China University of Geosciences, Beijing 100083*)

2(*State Key Laboratory of Space Weather, Center for Space Science and Applied Research,
Chinese Academy of Sciences, Beijing 100190*)

Abstract The three-dimensional numerical Magnetohydrodynamics (MHD) simulation is one of the methods to study the corona and solar wind. Comparison of the Polarization Brightness (PB) in the low corona between simulation results and observation is important to validate the MHD models. Because of the massive data and the complexity of the PB model, the computation will take too much time on single CPU (or core), and can not visualize the PB in nearly real time, so that it affects the validation efficiency of numerical modeling. A new parallelized model based on CPU/GPU with Compute Unified Device Architecture (CUDA) to convert MHD simulation density to PB is presented in this paper. Aiming at the most time-consuming part of the serial PB model, it modifies the model from serial process to parallel process based on CUDA to improve its computation efficiency. It shows that the new model can speed up 31.86 times than the serial model based on

* 国家重点实验室专项基金资助项目

2009-05-09 收到原稿, 2010-03-22 收到修定稿

E-mail: dkzhong@spaceweather.ac.cn

CPU, and is able to compare the simulation with observation in nearly real time.

Key words Coronal polarization brightness, Graphic Processing Unit (GPU), Compute Unified Device Architecture (CUDA)

1 引言

空间天气预报需要一个准确的太阳大气和行星际空间太阳风模型。太阳日冕和日球层的全球模式是空间天气数值预报的基础,也是解释各种相关观测的依据^[1]。三维磁流体力学(MHD)模拟是用来研究日冕和太阳风最常用的数值方法之一,其中将计算得到的日冕电子数密度转化为日冕偏振亮度(PB)是与观测数据对比的重要方法,也是验证空间天气预报数值模拟模型合理性的重要手段之一。

本文关于PB的计算,是以MHD数值模型产生的电子数密度作为输入参数,由PB的计算公式计算指定天平面的PB值。由于待转换电子数据网格密度、PB数据网格密度及计算PB的复杂性,PB计算隐含着大量的计算和存储需求。为了达到近实时转换日冕偏振亮度,提高数值模式验证效率的目的,改善PB计算性能十分关键。本文根据CPU/GPU计算环境特点,深入分析了PB的转换算法,利用CUDA编程技术,提出了一个基于CPU/GPU的PB并行计算模型,并与CPU串行模型相比较,速度提高了31.86倍。

2 PB 的计算模型

PB来源于日冕中自由电子对光球辐射的汤姆逊散射(Thomson Scattering),可以用来反演日冕的电子数密度,还可以用来检验数值模型与观测结果的符合程度。考虑到临边昏暗等因素,日冕的一个小体积元PB的计算公式如下^[2]:

$$I_t - I_r = I_0 \frac{N_e \pi \sigma}{2} \sin^2 \chi [(1 - \mu)A + \mu B], \quad (1)$$

$$A = \cos \Omega \sin^2 \Omega, \quad (2)$$

$$B = -\frac{1}{8} \left[1 - 3 \sin^2 \Omega - \frac{\cos^2 \Omega}{\sin \Omega} (1 + 3 \sin^2 \Omega) \cdot \ln \frac{1 + \sin \Omega}{\cos \Omega} \right]. \quad (3)$$

其中, I_t , I_r 为日冕中任意一点的切向和径向散射辐射分量; I_0 为日面中心亮度; N_e 为日冕电子数密度,

σ 为电子散射截面; Ω 为 OP 与从 P 到日面切线间的夹角(见图1); A, B 为 Ω 的函数; χ 为任一点的日心连线与视线方向的夹角; μ 为波长的经验函数,在本文中为常数0.63。

如图1所示,沿观测点和天平面上各点的连线,即视线,对式(1)进行积分,可以得到用于与日冕仪观测结果对比的天平面日冕偏振亮度图。

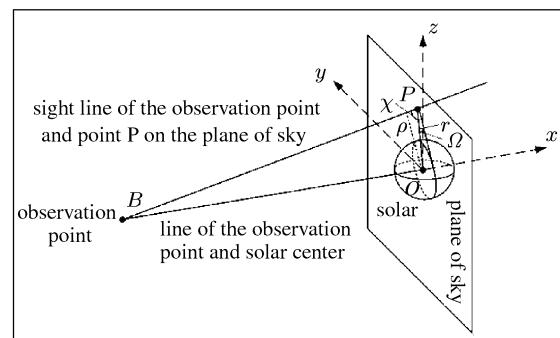


图1 PB 计算时的密度积分过程 (B 为成像仪观测点, P 为天平面上任意一点, O 为日心, ρ 为日心到视线的距离)

Fig.1 Density integral process of PB calculation

(B is the observation point, P is a point on the plane of sky, O is the solar center, and ρ is the distance from the solar center to the line of sight)

本文采用了文献[3-4]中规模为 $42 \times 42 \times 82$ (r, θ, φ) 的电子数密度数据作为实验数据(其中, $1 R_s < r < 23 R_s$), 经过计算将产生 321×321 ($-4 R_s \leq y \leq 4 R_s$, $-4 R_s \leq z \leq 4 R_s$) 的天平面PB值数据。在计算过程中,如图1所示,以日心 O 为原点, 观测点 B 与日心 O 的连线方向为 x 轴, 天平面所在截面为 Oyz 平面, 建立 $960 \times 320 \times 320$ ($-12 R_s \leq x \leq 12 R_s$, $-4 R_s \leq y \leq 4 R_s$, $-4 R_s \leq z \leq 4 R_s$) 的直角坐标系均匀网格。对天平面任意一点 P , 沿 BP 方向(视线方向)对电子偏振亮度,即式(1)累积求和,可以得到天平面上该点的偏振亮度值。

3 基于 CPU 的串行计算模型

针对本文的实验数据,在CPU上的串行计算模

型步骤如图 2 所示.

步骤 1 根据 MHD 数值模拟结果的电子数密度数据的大小, 分配计算时所需的内存空间.

步骤 2 将计算域内的电子数密度数据读入内存.

步骤 3 初始化电子数密度的球坐标网格信息, 根据成像仪在此坐标系下的位置, 确定天平面, 并建立 $960 \times 320 \times 320$ 的直角坐标系均匀网格.

步骤 4 根据 PB 计算公式, 沿视线方向进行偏振亮度积分求得天平面上各网格点的 PB 值. 在此过程中, 由于原始电子数密度的球坐标网格和由天平面确定的直角坐标网格不是一一对应的, 必须确定直角坐标网格各点在原球坐标中的位置, 进而插值计算直角坐标内视线上各点的电子数密度.

步骤 5 根据计算出来的结果, 绘制出日冕偏振亮度图.

根据上述 PB 串行计算流程, 在一台图形工作站 (配置 Intel® Xeon® CPU, E5405, 2.00GHz (2 CPUs, 8 Cores), 1 GB 内存) 和 Windows Visual Studio 2005 的实验环境下, 仅用 CPU 计算的结果列于表 1. 从表 1 可以看到, PB 值计算部分占程序总运行时间的 99.05%. 因此, 要达到近实时转换 PB 的目的, 压缩步骤 4 对 PB 值计算部分的时间是十分必要的. 由于在 PB 计算部分中沿观测点与天平面各点的视线方向上的积分计算过程是相互独立的, 即每个待计算的 PB 与其他待计算的值之间没有相关联系, 因此, 很适合利用并行计算方式对此问题进行求解以

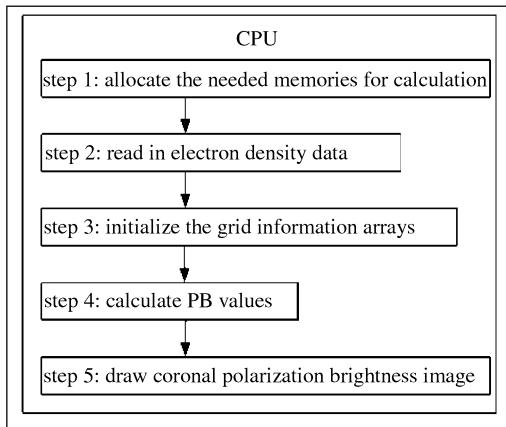


图 2 PB 计算的串行流程

Fig. 2 Serial process of PB calculation

表 1 PB 串行计算模型中各步骤耗时及占用总耗时的百分比

Table 1 Each step time and its their percentage of total time in PB serial calculation model

步骤	耗时/s	单步耗时占用总耗时的百分比/ (%)
分配内存空间	可忽略	0
读取数据	0.218	0.44
初始化网格信息	可忽略	0
PB 值计算	48.938	99.05
根据计算结果绘制日冕偏振亮度图	0.250	0.51

提高程序性能. 由于 GPU 在密集型计算和并行计算方面具有优势, 所以, 本文利用 CUDA 技术对 PB 计算进行高度并行化, 尝试为 MHD 数值计算结果的后处理提供一种可行的解决方案.

4 基于 GPU/CPU 的并行计算模型

4.1 基于 GPU 的 CUDA 技术

CUDA (Compute Unified Device Architecture) 是美国 NVIDIA 公司推出的一种基于 GPU 的并行计算体系架构, 它支持 C 编程, 以及 Windows 和 Linux 运行环境, 目前 CUDA 技术已经应用于很多领域 [13–15].

在 CUDA 架构下, 程序分为 host 端和 Device 端, Device 端的程序称为 Kernel 函数. 程序运行时, host 端程序将数据复制到 GPU 的内存, 再由 GPU 执行 Kernel 函数, 最后由 host 端程序将结果从 GPU 内存中取回 [15]. GPU 上执行的最小单位是 thread, 多个 thread 组成一个 block. 每个 block 中的 thread 可共享同一内存数据, 并可进行快速同步操作. 执行相同程序的多个 block 可组成一个 grid, 如图 3 所示 [15]. CUDA 可以在 N 个 thread 中执行相同的操作来实现数据级并行计算.

在 CUDA 架构下, GPU 上的内存模型有 global memory, local memory, texture memory, constant memory, shared memory 和 register, 如图 4 所示 [15]. 其中 global memory 提供的存储空间最大, 但操作最耗时. 例如, 从 global memory 中读取一个 float 类型

数据需要 400~600 个时钟周期, 而操作 shared memory 只需要 4 个时钟周期。各种内存模型耗时从多到少排序为 global memory > local memory > shared memory > constant memory > texture memory > register^[16]。

本文实验中所用的 Geforce 8800 GTX 显卡基于 G80 核心。该 GPU 拥有 16 个多处理器, 每个处理器上有 8 个流处理器, 总共有 128 个流处理器。不同的多处理器以 MIMD 模型执行不同的指令, 多处理器内部以 SIMD 模型执行计算。每个多处理器有 16 KB 的 shared memory, 8 KB 的 constant memory, 8 KB 的 texture memory 和 8192 个 32 bit 的 register^[16]; 每个多处理器最多可运行 768 个线程, 16 个多处理器最多可同时运行 12288 个线程。

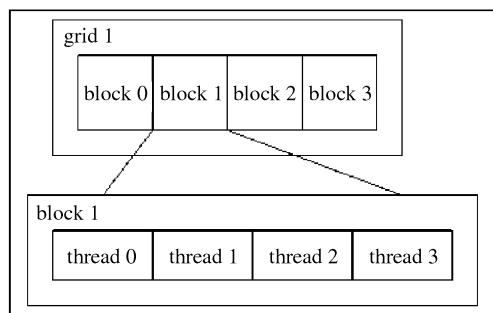


图 3 thread, block 和 grid 之间的关系

Fig. 3 Relationship among thread, block and grid

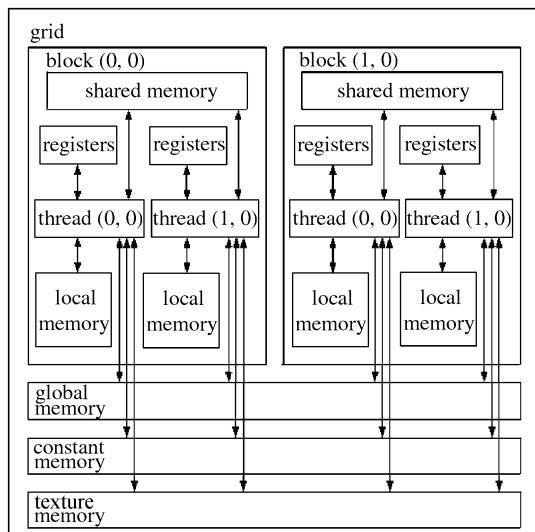


图 4 CUDA 架构中各内存模型之间的关系

Fig. 4 Relationship among the memory models in CUDA

4.2 并行计算模型

并行计算模型也在前述图形工作站上进行计算测试, 其并行计算部分由该工作站上配置的 NVIDIA Quadro FX 4600 GPU 显卡 (G80 核心架构, 12 个多处理器, 128 个并行流处理器, 760MB GDDR3 SDRAM) 来完成。

根据 PB 串行计算模型的计算结果分析和 CUDA 的编程构架, 考虑将 PB 值的计算部分放入 Kernel 函数中, 以实现程序的并行化。而在计算每个 PB 值时, 沿视线方向进行的积分过程都是相互独立的。因此, 可以利用多线程进行数据处理, 每个线程独立计算一个 PB 值。由于待计算 PB 值的数目远大于 GPU 可启用的线程数目, 因此每个线程需循环计算多个 PB 值。根据实验条件, 每个 block 设定的线程数目为 256 个, 这样可最大化地利用计算资源。block 的数量取决于待计算 PB 的数目与线程数目之比。

由于 global memory 存取耗时太多, 可以将部分相对独立的数据载入到 shared memory, 这样可缩减数据访问时间。在 PB 并行计算模型中, 可以放进 shared memory 中的数组包括两个坐标系下的网格信息, 即 $r, \theta, \varphi, dr, x, y, z$ 。在 CPU 与 GPU 间多次传输数据比较耗时, 因此可将这些数据打包进一个数组 Sdata, 以减少数据传输的次数。为了在程序中能准确地提取 Sdata 中各部分数据内容, 需要一个记录各部分数据内容长度的数组, 使得在 GPU 上计算时, 可以进行原始数据与打包后数据之间的索引转换, 并在调用 Kernel 函数前确定使用共享内存的大小。经过测算, 待放入 shared memory 的数据大小约 7 KB, 小于 GPU 提供的 16 KB, 因此并行方案可行。并行计算流程如图 5 所示, 表 2 给出了计算结果。与串行

表 2 PB 并行计算模型中各步骤耗时

Table 2 Each step time in PB parallel calculation model

步骤	耗时/(s)
分配内存空间	可忽略
读取数据	0.218
初始化网格信息	可忽略
PB 值计算	3.4335
根据计算结果绘制日冕偏振亮度图	0.250

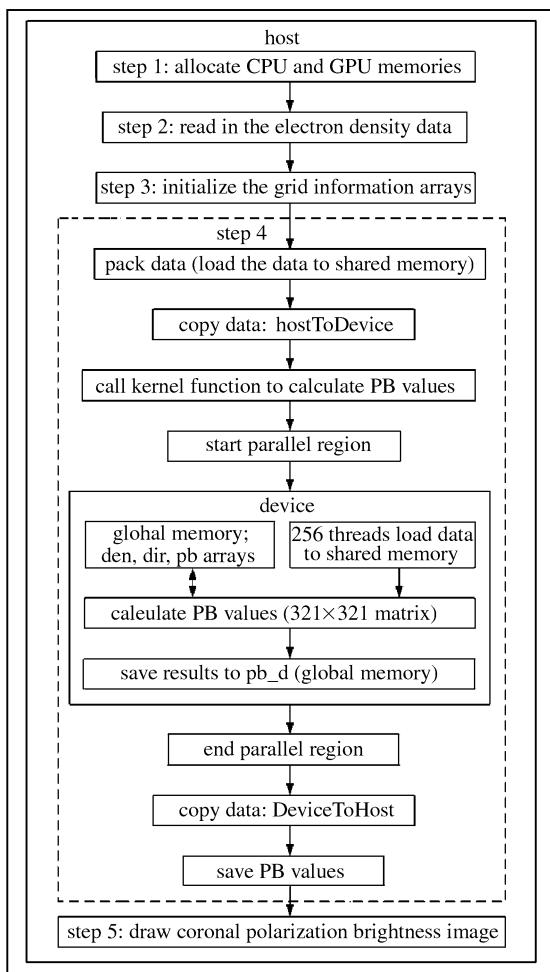


图 5 PB 的并行计算流程

Fig. 5 Parallel process of PB calculation

模型相比, 并行模型的 PB 值计算效率提高了 14.25 倍, 计算时间缩短到 3.4335 s.

4.3 并行模型的改进

进一步分析并行模型的具体计算内容, 可从以下两个方面对模型进行改进.

① 转换索引时用到的数据长度数组是只读的, 并且在计算过程中使用此数据的频率很高, 因此提高这部分数据的访问速度可以进一步优化模型. 在 CUDA 架构提供的内存模型中, constant memory 更适合存放此类数据, 本文采用的优化策略是将该数组从 shared memory 迁移到 constant memory, 进一步提高 Kernel 函数访问该数组的效率.

② CUDA 编译器在某些情况下会插入一些转换指令, 增加额外的执行周期^[15]. 例如, 在需要双精度输入参数的函数中传入单精度参数时即会出现这种

情况. 因此, 在保证精度的条件下, 在计算模型中采用 CUDA 编译器提供的函数, 例如 $\sin()$, $\log()$, 来进一步提高模型的运算效率.

经过上述改进后的 PB 并行计算模型的运行结果列于表 3. 与原始并行模型相比, 改进后并行模型 PB 值的计算效率提高了 2.24 倍, 计算时间缩短到 1.536 s, 进一步提高了模型的计算性能.

如表 4 所示, 最终的计算结果表明, 与串行计算模型相比, 最后的并行模型 PB 值计算效率提高了

表 3 改进的 PB 并行计算模型中各步骤耗时

Table 3 Each step time in PB parallel improved calculation model

步骤	耗时/(s)
分配内存空间	可忽略
读取数据	0.218
初始化网格信息	可忽略
PB 值计算	1.536
根据计算结果绘制日冕偏振亮度图	0.250

表 4 三种计算模型的 PB 值计算时间及并行模型的加速比

Table 4 PB calculation time of three models and speedup of parallel models

	PB 值计算时间/(s)	加速比
串行模型	48.938	
并行模型	3.4335	14.25
改进后的并行模型	1.536	31.86

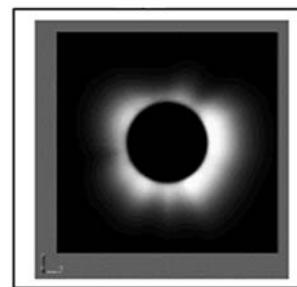


图 6 实验数据计算得到的日冕偏振亮度

Fig. 6 Coronal polarization brightness image calculated by experimental data

31.86 倍, 计算时间由原来的 48.938 s 缩短到 1.536 s, 并行模型的加速效果非常明显. 图 6 给出了计算得到的日冕偏振亮度结果.

5 结论

利用 CUDA 技术建立了基于 CPU/GPU 的日冕偏振亮度并行计算模型, 并且根据 CUDA 架构的性能特点, 对并行模型从两个方面进行改进. 实验结果表明, 与串行模型比较, PB 值计算时间由原来的 48.936 s 缩短到 1.536 s, 运行速度提高了 31.86 倍.

随着数值空间天气模式的深入研究和计算条件(计算速度和存储空间)的不断进步, MHD 数值计算产生的数据也在不断膨胀; 同时, 对这些数据进行后处理及可视化的实时性要求也越来越高, 仍然采用以往串行后处理和可视化的方式已经不能满足要求. 本文为解决这个问题做了初步尝试, 在从电子数密度到 PB 的转换和可视化过程中基本达到近实时的要求. 由此可以预见, 基于 GPU 的并行计算模型将在 MHD 数值计算结果的后处理和可视化中具有广泛的应用前景.

参考文献

- [1] Xiang Changqing, Feng Xueshang, Fan Quanlin, Yao Jiusheng. An observation-based model of solar wind background [J]. *Chin. J. Space Sci.*, 2006, **26**(3):161-166. In Chinese (向长青, 冯学尚, 范全林, 姚久胜. 一个基于观测的太阳风背景模型 [J]. 空间科学学报, 2006, **26**(3):161-166)
- [2] Billings D E. A Guide to the Solar Corona [M]. New York: Academic Press, 1966
- [3] Feng X S, Zhou Y F, Wu S T. A novel numerical implementation for solar wind modeling by the modified conservation element/solution element method [J]. *Astrophys. J.*, 2007, **655**:1110-1126
- [4] Hu Y Q, Feng X S, Wu S T, Song W B. Three-dimensional MHD modeling of the global corona throughout solar cycle 23 [J]. *J. Geophys. Res.*, 2008, **113**: A03106
- [5] Manavski S A. Cuda compatible GPU as an efficient hardware accelerator for AES cryptography [G]. //Proceedings of IEEE International Conference on Signal Processing and Communication, 2007, **11**:65-68
- [6] Claudio Albanese, Alicia Vidler. Dynamic conditioning and credit correlation baskets [M]. Germany: MPRA Paper from University Library of Munich, 2008
- [7] Michael M Heck. 3D visualization for oil and gas evolves [M/OL]. Mercury Computer System, Inc. <http://www.hpcwire.com/hpc/948371.html>, 2006
- [8] Jeong Won Ki, Thomas Fletcher P, Tao Ran, Whitaker Ross T. Interactive visualization of volumetric White Matter connectivity in DT-MRI using a parallel-hardware Hamilton-Jacobi solver [G]. //IEEE Conference on Visualization, 2007
- [9] Holger Scherl, Benjamin Keck, Markus Kowarschik, Joachim Hornegger. Fast GPU-based CT reconstruction using the Common Unified Device Architecture (CUDA) [R]. Nuclear Science Symposium Conference Record, 2007
- [10] Maud Marchal, Jeremie Allard, Christian Duriez, Stephane Cotin. Towards a framework for assessing deformable models in medical simulation [G]. //Proceedings of the International Symposium on Computational Models for Biomedical Simulation (ISBMS), 2008
- [11] Michalakes J, Vachharajani M. GPU acceleration of numerical weather prediction [J]. *Parallel Proc. World Sci.*, 2008, **18**(4):531-548
- [12] Andrew Corrigan, John Wallin, Matej Vesnjak. Visualization of meshless simulations using fourier volume rendering [G]. //Progress on Meshless Methods, Computational Methods in Applied Sciences Series, 2009, **11**:291-305
- [13] Tsuyoshi Hamada, Keigo Nitadori, Tomonari Masada, Makoto Taiji. 50.5 Mflops/dollar and 8.5 Tflops Cosmological N-body Simulation on a GPU Cluster [R]. ACM/IEEE Conference, 2008
- [14] Michael C Schatz, Cole Trapnell. Fast exact string matching on the GPU [R]. //Center for Bioinformatics and Computational Biology, 2007
- [15] NVIDIA CUDA Compute Unified Device Architecture Programming Guide 1.1 [M/OL]. NVIDIA Corporation, http://developer.download.nvidia.com/compute/cuda/1_1/NVIDIA_CUDA_Programming_Guide_1.1.pdf. 2007
- [16] Ryoo S, Rodrigues C I, Baghsorkhi S S, Stone S S, Kirk D B, Hwu W. Optimization principles and application performance evaluation of a multithreaded GPU using CUDA [R]. //Proceedings of ACM PPoPP'08, 2008