

基于 GPU 的电力系统并行潮流计算的实现

夏俊峰¹, 杨帆¹, 李静², 郑秀玉¹

(1. 武汉大学电气工程学院, 湖北 武汉 430072; 2. 武汉国测科技股份有限公司, 湖北 武汉 430223)

摘要: 在研究 GPU 通用计算方法和潮流计算算法的基础上, 针对 GPU 计算密集、高度并行化等特点, 对潮流计算牛顿法进行了适当的简化, 并应用统一计算设备架构 (Compute Unified Device Architecture, CUDA) 的开发平台, 提出了一种基于 GPU 的并行潮流计算程序的设计方法。仿真计算结果表明此算法可行, 并具有较高的计算效率, 为电力系统并行潮流计算的研究提供了一种可行的方法。

关键词: 潮流计算; 并行计算; GPU 通用计算; 统一计算设备架构; 牛顿法

Implementation of parallel power flow calculation based on GPU

XIA Jun-feng¹, YANG Fan¹, LI Jing², ZHENG Xiu-yu¹

(1. School of Electrical Engineering, Wuhan University, Wuhan 430072, China;

2. Wuhan GUOCE S&T CO.,LTD, Wuhan 430223, China)

Abstract: A design method for GPU-based parallel power flow calculation program is presented. This method is based on the study of the power flow calculation methods and the general-purpose GPU computing algorithm, appropriately simplifies Newton power flow calculation method to adapt to GPU's features of compute-intensive, highly parallel and so on, and applies the Compute Unified Device Architecture (CUDA) Development Platform. The simulation results show that this algorithm is feasible and has a high computational efficiency. It provides a viable approach for the trend of power system in parallel computing research.

Key words: power flow calculation; parallel computation; general-purpose GPU; compute unified device architecture; Newton method

中图分类号: TM74 文献标识码: A 文章编号: 1674-3415(2010)18-0100-04

0 引言

电力系统潮流计算是电力系统规划、运行的基本研究方法。随着现代电力系统的不断发展, 电网规模和电网设备数日益增大, 其潮流计算的计算量与计算复杂度急剧增加。传统的串行潮流计算算法在计算速度上已经不能满足大规模电网潮流实时计算的要求, 随着计算机技术与并行计算技术的不断发展、成熟, 大规模电网潮流计算的并行算法研究已获得了长足的发展。

目前的并行潮流计算^[1-2], 在技术层面上主要采用分布式计算、并行机或多线程等并行处理技术, 在取得较好效果的同时还存在下述不足: 对于大规模电网, 庞大的节点规模导致分布式计算机之间、多线程之间巨大的通讯消耗, 而大多数并行机

难以承受; 且并行机价格昂贵、使用针对性强, 主要面对高端用户, 其管理和使用相对复杂, 难于普及; 多线程技术是在多核 CPU 或单核 CPU 上用串行来模拟并行, 速度提高有限。

近几年来, 图形处理器 GPU 发展迅猛, 已发展成为一种可编程、高度并行化、多线程、多核的处理器, 具有杰出的计算效率和极高的存储带宽, 为通用计算提供了良好的并行计算平台。通过 GPU 进行并行算法加速, 是解决大规模电网潮流并行计算的一种可行的低成本方法。

本文在 GPU 通用计算方法和电力系统潮流计算牛顿法的基础上, 提出一种改进的基于 GPU 加速的并行潮流计算算法的实现方法。针对 GPU 的计算密集、高度并行化等特点, 对牛顿法进行适当的简化, 并通过统一计算设备架构 (Compute Unified Device Architecture, CUDA) 提供的 API 和开发平台编程实现该算法, 提高了并行潮流计算的运行速

基金项目: 国家电网公司项目资助 (SGKJJSKFC 2008) 449 号)

度, 为普通用户研究并行潮流计算提供了一种可行的方法。

1 GPU 通用编程模型及 CUDA

1.1 可编程 GPU 处理器

CPU 与 GPU 处理器各有专长, 某种意义上说 CPU 是万能的, 它几乎可以处理任何事情, 适合逻辑运算, 但由于深度流水作业架构的特性、以及浮点运算能力的限制, 它处理一些计算密集型任务时的效率很低。早期显卡只是为屏幕上显示像素提供一个缓存, 所有的图形处理都是由 CPU 单独完成。近年来 GPU 的飞速发展, 一步步地将那些不适合 CPU 处理、或 CPU 算不动的任务转移过来, 从而消除程序运算时的瓶颈, 大幅提升计算机执行效能, 以更小的代价 (成本和功耗) 实现更强大的性能。GPU 的性能更新非常迅速, 平均一年性能提升一倍, 如表 1 所示 2008 年度 GPU 与 CPU 价格相当的情况下, GPU 的计算能力已经远远高于 CPU 的计算能力。

表 1 GPU 与 CPU 计算能力比较

计算能力	Intel Core i7920	Geforce GTX 285	运算能力比较
核心数	4 个	240 个	GPU 60 倍
浮点运算	60 GFLOPS	1000 GFLOPS	GPU 快 16.67 倍
存储带宽	20 GB/s	159 GB/s	GPU 快 7.95 倍

1.2 基于 GPU 的通用计算

基于 GPU 的通用计算 (General-Purpose GPU, GPGPU), 是研究如何把 GPU 具有的强大计算能力, 应用于除图形处理以外的通用计算领域。目前 GPGPU 的问题在于, GPU 处理器是专为图形处理设计的处理器, 具有自己的存储单元, 在数据存取方式上存在一定的特殊性, 而且通常 GPU 的软件开发采用的非标准的编程模式、编程环境和体系结构, 研究人员在研究并行算法的同时还必须研究算法在 GPU 运行结构上的优化^[3]。

直到现在, 获得 GPU 中的所有计算能力并将其有效用于非图像应用程序中仍然是一个难题, 正如前面所说的困难, 迫切需要一种新颖的硬件和编程模型, 让 GPU 的计算能力得以完全发挥, 并将 GPU 作为一种真正通用的数据并行计算设备。CUDA 是 NVIDIA 公司的 GPGPU 模型, 它使用 C 语言为基础, 可以直接以大多数人熟悉的 C 语言写出在显示芯片上执行的程序, 而不需要去学习特定显示芯片的指令或是特殊结构。CUDA 作为一种新型的硬件和软件架构, 用于将 GPU 作为数据并行计算设备,

在 GPU 上进行计算任务的发放和管理, 而无需将其映射到图像 API。

1.3 CUDA 编程模型

CUDA 软件栈包含多个层, 如图 1 所示: 设备驱动程序、应用程序编程接口 (API) 及其运行库、两个较高级别的通用数学库^[4]。

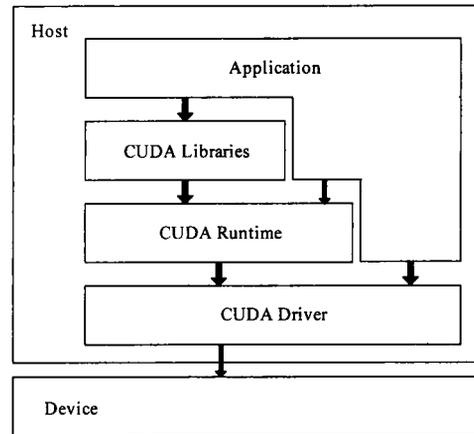


图 1 统一计算设备架构软件栈

Fig.1 Software stack of CUDA

在 CUDA 架构下, 一个程序分为两个部份: host 端和 device 端。Host 端是指在 CPU 上执行的部份, 而 device 端则是在显示芯片上执行的部份。Device 端的程序又称为内核 (kernel)。通常 host 端程序会将数据准备好后, 复制到显卡的内存中, 再由显示芯片执行 device 端程序, 完成后再由 host 端程序将结果从显卡的内存中取回。由于 CPU 只能通过 PCI Express 接口存取显卡内存, PCI Express x16 接口的理论带宽是双向各 4 GB/s, 速度较慢, 因此不能频繁进行这类运算操作, 以免降低效率。

CUDA 允许程序员定义称为内核 (kernel) 的 C 语言函数, 与普通的 C 语言函数只执行一次的方式不同, 在调用此类函数时, 它将由 n 个不同的 CUDA 线程并行执行 n 次。通过 CUDA 编程时, 程序线程如图 2 所示, 将 GPU 看作可以并行执行非常多个线程的计算设备, 作为主 CPU 的协处理器来运作, 把在主机上运行的应用程序中数据并行的、计算密集的部分卸载到此设备上。CUDA 在执行的时候是让 host 里面的一个一个的内核 (kernel) 按照线程网格 (Grid) 的概念在显卡处理器 GPU 上执行。每一个线程网格又可以包含多个线程块 (block), 每一个线程块中又可以包含多个线程 (thread)。一个线程块内的线程可彼此协作, 通过一些共享存储器来共享数据, 并同步其执行来协调存储器访问。

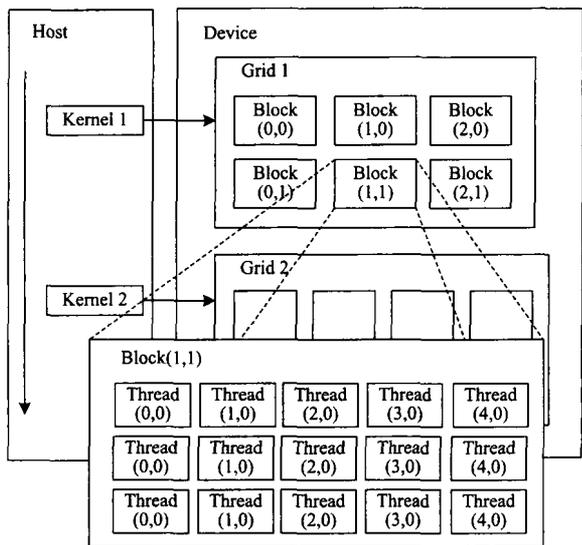


图 2 CUDA 程序线程结构
Fig.2 Thread structure of CUDA

2 并行潮流算法

潮流计算是研究电力系统运行状态的基本计算，多年来为提高潮流计算的速度，研究人员做了很多工作，提出了多种实用算法，这些算法大多都是对大规模电网进行区域划分进行粗粒度并行计算，并行度不高。而 GPU 处理器的特点适用计算密集型任务，能够实现细粒度的并行计算。因此这些算法不能直接适用于 GPU 并行计算，需针对 GPU 处理器和 CUDA 架构的特点进行优化。

牛顿-拉夫逊法具有适应性好、收敛快的优点，在电力系统潮流计算中有着广泛的应用。但是在网络节点较多时其计算量急剧增大，计算速度慢。牛顿法中大量的计算在于大规模雅克比矩阵单元的计算。在保证计算收敛的条件下，可以对牛顿法的迭代计算过程进行合理的简化，将迭代过程中修正方程的系数矩阵转化为常数矩阵。

2.1 简化牛顿法的推导

牛顿法的迭代与修正方程式^[5]表示为

$$F(X^{(k)}) = -J^{(k)}\Delta X^{(k)} \quad (1)$$

$$X^{(k+1)} = X^{(k)} + \Delta X^{(k)} \quad (2)$$

式中： $X^{(k)}$ 和 $\Delta X^{(k)}$ 分别是由 n 个变量和修正量组成的 n 维列向量； $F(X^{(k)})$ 是由 n 个多元函数组成的 n 维列向量；雅克比矩阵 J 是 $n \times n$ 阶方阵，它的第 i 、 j 个元素 $J_{ij} = \partial f_i / \partial x_j$ 是第 i 个函数 $f_i(x_1, x_2, \dots, x_n)$ 对第 j 个变量的偏导数。迭代过程中，每迭代一次就必须计算一次雅克比矩阵 J ，当网络规模大节点较多时，计算量非常大。

简化牛顿法，将迭代过程的修正方程的系数矩阵转化为常数矩阵，减少牛顿法迭代过程中计算雅

克比矩阵的计算量，来提高潮流计算的速度。简化牛顿法只在第一次计算修正量时计算雅克比矩阵，在以后的迭代过程中并不重新计算雅克比矩阵，其几何意义如图 3 所示。函数 $y = f(x)$ 如图中曲线， $f(x) = 0$ 的解相当于曲线与 x 轴的交点。过 $[x^{(0)}, y^{(0)} = f(x^{(0)})]$ 点作一切线，如果第 k 次迭代中得到 $x^{(k)}$ ，过 $[x^{(k)}, y^{(k)} = f(x^{(k)})]$ 点作初始切线的平行线，此平行线与 x 轴的交点便确定了下一个近似解 $x^{(k+1)}$ ，重复这个过程直到与真值误差小于给定的小正数。

简化牛顿法所做的简化只涉及到解题过程，而收敛条件的校验依然是以精确的模型为依据，其计算结果的精度是不受影响的；其收敛性由二阶收敛变为接近线性收敛，且对某些含类似配电支路的网络难以收敛或收敛精度降低，故适用范围较牛顿法窄。可以通过计算雅克比矩阵时保留泰勒级数的二阶项，改善该算法的收敛性。

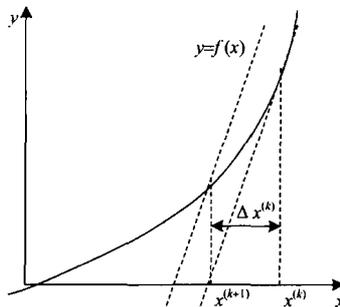


图 3 简化牛顿法的几何意义
Fig.3 Geometric meaning of simplified Newton method

2.2 并行计算算法实现

按照 CUDA 编程的思想，将简化牛顿法迭代过程中修正量计算部分定义为内核函数，在 GPU 设备上由多线程执行。本文采用雅克比逆矩阵的方法求解修正方程，对于一个 n 节点的网络，其雅克比矩阵为 $2(n-1)$ 阶的方阵，修正量等于 $2(n-1)$ 阶方阵与 $2(n-1)$ 阶列向量的乘积，每个内核函数线程处理一个数值乘法运算。

雅克比矩阵是一稀疏矩阵，牛顿法中为减少计算量通常应用稀疏技术求解。本文旨在实现所提方法，在求解修正方程时都采用求雅克比逆矩阵法，而实际应用中可以进一步优化改进。

并行算法程序分为 CPU 端和 GPU 端两个部分，GPU 设备上执行解修正方程的内核函数，内核函数计算所需的雅克比逆矩阵、第 k 次的迭代值及计算结果均由 CPU 端进行计算，GPU 与 CPU 间的数据通讯通过设备主内存与显存间的内存拷贝实现，程序流程图如图 4 所示。

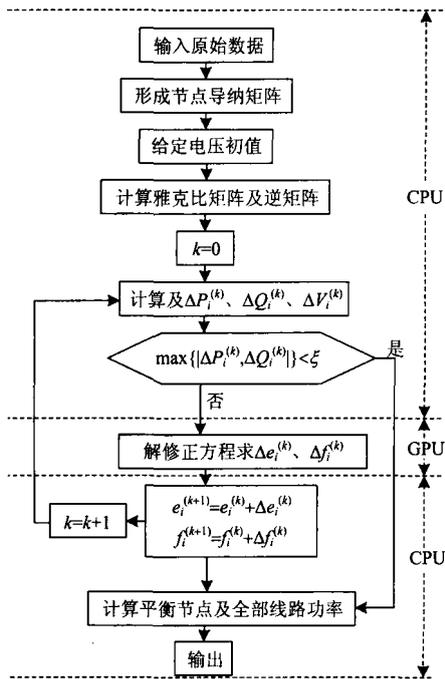


图4 计算程序流程图

Fig.4 Computer program flowchart

在 CUDA 程序中, 线程是分块的, 一个线程块最多只能有 512 个线程。一个线程块中的线程, 具有一个共享的内存, 并可进行同步, 不同线程块之间的线程则不行。在本文的程序中, 不需要进行线程间的同步动作, 因此可以使用多个线程块来增加线程的数目, 取每个线程块 256 个线程, 线程块的个数为 $(2(n-1))^2/256+1$ 。简化牛顿法其迭代过程中雅可比矩阵和其逆矩阵是不变的, 迭代计算时内核函数不用重复拷贝雅可比逆矩阵, 而只需拷贝第 k 次迭代值和计算结果, 避免了主内存与显存间的大量数据通信的消耗。

3 仿真计算结果与分析

本文通过编程, 分别实现了基于 CPU 计算及 GPU 并行计算的潮流计算程序, 在计算机上进行多次重复计算系统潮流, 来验证基于 GPU 并行潮流计算的计算效率。

测试平台采用 intel E5200 3.33GHz CPU、2G 内存、Geforce 9600 GSO 显卡、Windows XP SP3 操作系统。采用 IEEE 标准 4-300 节点系统数据, 计算雅可比矩阵时只保留泰勒级数的一阶项, 算法单次收敛的迭代次数和 100 次计算所需总时间 (单位 ms) 如表 2 所示, 计算精度为 10^{-5} 。

分析表中数据可见, 简化牛顿法的收敛过程受初值和初始雅可比矩阵的影响, 所需的迭代次数较多, 可以通过计算雅可比矩阵时保留泰勒级数的二

表 2 GPU 和 CPU 并行计算仿真结果
Tab.2 Simulation results of CPU calculation and GPU parallel calculation

节点数	迭代次数	CPU 运行时间	GPU 运行时间	加速比
4	9	2	4 204	0
6	5	5	4 211	0
9	7	9	4 623	0
14	4	22	5 255	0
30	5	116	5 302	0.022
39	10	265	5 560	0.048
57	12	1 123	7 192	0.156
118	8	7 650	9 410	0.813
300	20	119 637	24 322	4.919

阶项来改善该算法的收敛性。表中数据显示, 当网络节点较少时, 基于 GPU 的并行计算程序所需的计算时间远远大于 CPU 计算程序, 只有节点数等于 300 时才能获得较好的加速比。这是因为:

(1) 因显卡数据总线带宽限制, 主内存和显卡内存间访问速度较慢。基于 GPU 并行运算程序需要在主内存和显卡内存间拷贝数据, 这个过程消耗了大量的计算时间。这也是本文提出对牛顿法迭代过程进行简化以减少雅可比矩阵数据拷贝的原因。

(2) 显示芯片并行计算的内存存取延迟的问题。CPU 通常使用缓存来减少存取主内存的次数, 以避免内存延迟影响到执行效率。显示芯片则多半没有缓存, 而是通过并行化执行的方式来隐藏内存延迟, 即当第一个线程需要等待内存读取结果时, 则开始执行第二个线程, 依此类推。只有可以大量并行化的问题, 才能有效隐藏内存的延迟, 并有效利用显示芯片上的大量执行单元。

可见只有在大规模节点网络的计算中才能体现基于 GPU 并行计算的效率, 网络的节点越多基于 GPU 并行计算的效率越高。

4 结论

本文在研究 GPU 通用计算方法和潮流计算算法的基础上, 提出了适用于 GPU 并行处理的简化牛顿法, 并探讨了 CUDA 架构下并行潮流计算程序设计的方法。通过仿真计算结果验证了基于 GPU 并行潮流计算算法可行性, 且具有较高的计算效率, 为基于 GPU 计算的电力系统并行潮流计算的研究提供了一种可行的思路。当然, 要使该算法真正实际应用起来, 特别是在大系统的分析计算中, 还需要做大量的工作。

(下转第 110 页 continued on page 110)

LEI Yuan-chao, CHEN Chun-gen, SHEN Jun, et al. Research on control method of maximum power point tracking for photovoltaic power system[J]. Advanced Technology of Electrical Engineering and Energy, 2004, 23 (3): 76-80.

[3] 王环, 金新民. 光伏并网逆变器最大功率点的跟踪控制[J]. 电子产品世界, 2004 (4): 88-90.
WANG Huan, JIN Xin-min. The MPPT method for photovoltaic grid-connected inverter[J]. Electronic Engineering & Product World, 2004 (4): 88-90.

[4] 李然, 孙晓, 陈桂兰. 光伏发电系统最大功率点跟踪控制[J]. 电子技术应用, 2001 (8): 33-35.

[5] 李炜, 朱新坚. 光伏系统最大功率点跟踪控制仿真模型[J]. 计算机仿真, 2006, 23 (6): 239-243.
LI Wei, ZHU Xin-jian. The maximum power point tracking control of a photovoltaic power system[J]. Computer Simulation, 2006, 23 (6): 239-243.

[6] XIAO Wei-dong, Dunford W D. A modified adaptive hill climbing MPPT method for photovoltaic power systems[C]. //2004 35th Annual IEEE Power Electronics Specialists Conference. Aachen(Germany): 2004 : 1957-1963.

[7] 迟永宁, 李群英, 李琰, 等. 大规模风电并网引起的电力系统运行与稳定问题及对策[J]. 电力设备, 2008, 9 (11): 16-19.

CHI Yong-ning, LI Qun-ying, LI Yan, et al. Power system operation and stability problems caused by integration of large scale wind power and corresponding solutions[J]. Electrical Equipment, 2008, 9 (11): 16-19.

[8] 关宏亮, 赵海翔, 王伟胜, 等. 风电机组低电压穿越功能及其应用[J]. 电工技术学报, 2007, 22 (10): 173-177.
GUAN Hong-liang, ZHAO Hai-xiang, WANG Wei-sheng, et al. LVRT capability of wind turbine generator and its application[J]. Transactions of China Electrotechnical Society, 2007, 22 (10): 173-177.

收稿日期: 2009-10-16; 修回日期: 2009-12-09

作者简介:

杨德州 (1968-), 男, 硕士, 高级工程师, 副总工程师, 研究方向为电力系统一次规划、优质电源供电、无功与谐波抑制等方面的内容;

王利平 (1980-), 男, 硕士, 工程师, 研究方向为电力系统一次规划、电能质量监控、无功及其谐波抑制与补偿、大型分布式电源并网特性等方面; E-mail: leepingwang@tom.com

张军 (1976-), 男, 硕士, 副高级工程师, 研究方向为电力系统一次规划、电能质量监控、新能源并网特性等方面。

(上接第 103 页 continued from page 103)

参考文献

[1] 薛巍, 舒继武, 等. 电力系统潮流并行算法的研究进展[J]. 清华大学学报: 自然科学版, 2002, 42 (9): 1192-1195.
XUE Wei, SHU Ji-wu, et al. Advance of parallel algorithms for power flow simulation[J]. Journal of Tsinghua University: Science and Technology, 2002, 42 (9): 1192-1195.

[2] 吉兴全, 王成山, 等. 电力系统并行计算方法比较研究[J]. 电网技术, 2003, 27 (4): 22-26.
JI Xing-quan, WANG Cheng-shan, et al. A comparative study on parallel processing applied power system[J]. Power System Technology, 2003, 27 (4): 22-26.

[3] 张浩, 李利军, 等. GPU 的通用计算应用研究[J]. 计算机与数字工程, 2005, 33 (12): 60-62.
ZHANG Hao, LI Li-jun, et al. General-purpose computation on graphics processors[J]. Computer &

Digital Engineering, 2005, 33 (12): 60-62.

[4] 钱悦. 图形处理器 CUDA 编程模型的应用研究[J]. 计算机与数字工程, 2008, 36 (12): 177-180.
QIAN Yue. Compute unified device architecture on graphics processors[J]. Computer & Digital Engineering, 2008, 36 (12): 177-180.

[5] 何仰赞, 温增银. 电力系统分析[M]. 武汉: 华中科技大学出版社, 2002.

收稿日期: 2009-08-03; 修回日期: 2009-12-24

作者简介:

夏俊峰 (1979-), 男, 助理工程师, 硕士, 研究方向为电力系统检测与控制、电力系统测控技术; E-mail: jstormx@yeah.net

杨帆 (1982-), 男, 博士生, 研究方向为电力系统运行与控制;

李静 (1979-), 女, 工程师, 硕士, 研究方向为电力系统运行与控制。