

GPU 架构下基于经验模态分解的地震瞬时属性 并行提取算法的研究

曹晓初 金 弟 王宗仁 王启迪

(杭州地质研究院 杭州 310023)

摘 要 从地震勘探资料中提取地震瞬时属性具有十分重要的意义,而基于信号局部特征的经验模态分解为非线性非稳定信号提供了一种全新的瞬时属性提取方法。对经验模态分解算法在 GPU 架构上的并行处理实现进行了分析和研究。通过实验对比测试表明, GPU 架构下的算法运行效率较 CPU 具有明显优势。在测试数据中, GPU 加速比最高达到了 8.66 倍。

关键词 经验模态分解, 瞬时属性, 本征模函数, 并行处理, GPU

中图分类号 TP311.1 文献标识码 A

Research on the Seismic Instantaneous Attributes of Empirical Mode Decomposition Parallel Extraction Algorithm Based on GPU

CAO Xiao-chu JIN Di WANG Zong-ren WANG Qi-di

(PetroChina Hangzhou Research Institute of Geology, Hangzhou 310023, China)

Abstract It is significant that instantaneous attributes have been extracted from seismic data. Empirical mode decomposition based on the local characteristics provides a new analysis method suitable for nonlinear and non-stationary signals. EMD parallel algorithm has been analyzed and researched on GPU. Computation efficiency in GPU has substantial advantage of that in CPU by comparative experiment, and speedup of GPU is up to 8.66 compared to CPU among the test data in this article.

Keywords Empirical mode decomposition, Instantaneous attribute, Intrinsic mode function, Parallel process, GPU

1 前言

随着油气田的大规模开发, 未被发现的油气藏大多储量分散、层薄、埋藏深、孔渗条件复杂且难于被发现, 为了找到这些存储复杂的隐蔽性油气藏, 必须对采集到的信号做更准确更精细的处理, 以获得更详尽的地层信息。从地震勘探资料中提取地震瞬时属性具有十分重要的意义。地震瞬时属性通常包括瞬时振幅、瞬时相位和瞬时频率, 它们不仅可以用来研究地球介质的构造、岩性, 还可以反演介质的品质因子等参数^[1]。

勘探密度和精度的提高, 导致地震数据体的数据量快速上升, 一般数据多达几十 GB, 甚至几百 GB。而地震叠前数据的数据量较叠后数据常常高出数十倍, 甚至数百倍, 这使得传统 CPU 架构下地震信号属性的提取将变得更加耗时。目前 CPU 主频几乎已经达到了 CPU 工艺的极限, 并且随着主频的提高, 系统功耗不断上升, 成为单核 CPU 的主要障碍。GPU 的出现为大规模多线程并行处理地震数据提供一种技术实现。

如何快速高效地实现大规模地震数据的并行处理已成为亟待解决的重大课题之一。值得关注的是 GPU 计算能力的飞速发展及其应用, 特别是 GPU 并行处理技术的形成与发展, 为地震数据的并行处理提供了平台, 文献^[2-4]给出了一

种基于 GPU 的并行处理方法、希尔伯特-黄变换的实现方式等。凭借 CUDA 技术, 开发人员能够利用 GPU 解决复杂的并行计算难题, 开发出前所未有的高性能的应用软件。

2 GPU 技术

2.1 GPU 历史及发展现状

20 世纪 80 年代初期出现 GE(Geometry Engine)为标志的图形处理器, GE 的出现使得计算机图形学的发展进入图形处理器引导发展的年代。80—90 年代, GE 及其图形处理器功能不断增强和完善, 图形处理功能逐渐从 CPU 向 GPU 转移。

1999 年 NVIDIA 推出具有标志意义的 GeForce 图形处理器, 第一次在图形芯片上实现了 3D 几何变换和光照计算。此后 GPU 进入高速发展期, 平均每隔 6 个月就出现性能翻番的 GPU。

2002 年 ATI 发布的 Radeon 9700 和 2003 年 NVIDIA GeForce FX 的可编程 GPU 的出现, 使 GPU 可用于科学计算。2006 年 NVIDIA 与 ATI 分别推出了 CUDA(Computer Unified Device Architecture, 统一计算架构)编程环境和 CTM(Close To the Metal)编程环境, GPU 开始进入通用科学计算的领域^[5,6]。

2.2 GPU 与 CPU

从 CPU 和 GPU 微结构示意图上可以看出, CPU 中大量

的晶体管用作高速缓存、逻辑控制单元,只有少量用于计算单元(ALU)。而 GPU 则把更多的晶体管用在计算单元,只有少量晶体管做高速缓存和逻辑控制单元。CPU 设计的目的是尽可能串行高效地完成一件事,而 GPU 设计的目的则是尽可能高效并行地去做一件事。

CPU 擅长分支预测和逻辑判断,对满足条件的数据进行计算,在完成上一步的前提下继续下一步工作;而对矩阵运算、地震资料处理中的叠前时间偏移等 GPU 更适合。CPU 在操作系统、系统软件、应用程序、通用计算和系统控制等领域功能强大,而 GPU 在密集计算方面优势明显。

GPU 和 CPU 协同工作, GPU 处理大量的并行处理, CPU 处理操作系统和指令的逻辑控制。两者的协同比以往单纯 CPU 运算高出几十倍、几百倍甚至上千倍,可以使得 PC 和工作站具有超级计算的能力。

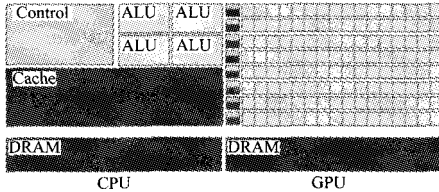


图1 CPU与GPU微结构示意图

CPU 在分支处理以及随机内存读取方面有优势,处理串行工作的能力较强; GPU 由于其特殊的核心设计,在处理大量有浮点运算的并行计算时能力较强。所以异构计算一般使用 CPU 做串行工作, GPU 主管并行运算。CPU 控制主程序的复杂流程,而将需要批量处理的向量数据通过高速总线传输给 GPU,由 GPU 实现快速并行计算。

3 GPU 架构下的经验模态分解

傅里叶变换和小波变换在处理非线性非平稳的信号时,均存在一定的缺陷。1996 年美国学者 N. E. Huang 提出一种全新的经验模态分解 (Empirical Mode Decomposition, EMD) 时频分析方法^[7]。和传统的非线性非稳定处理方法相比, EMD 方法能够很好地表述出信号的局部特征,从信号自身进行分解而不需要核函数,具有高分辨率和自适应性等。

3.1 经验模态分解(EMD)

从本质上讲,经验模态分解是对一个信号系列进行平稳化处理,将信号中不同尺度的波动或趋势逐级分解,产生一系列具有不同特征尺度的数据序列,每一个序列为一个本征模函数(Intrinsic Mode Function, IMF),最低频率的 IMF 分量通常情况下代表原始信号的趋势或均值。

信号 $x(t)$ 进行 EMD 分解,产生 n 个 IMF 分量 $c_i(t)$ 和趋势项 $r_n(t)$ 。即

$$x(t) = \sum_{i=1}^n c_i(t) + r_n(t) \quad (1)$$

每个 IMF 分量可以看成近似单一模态的平稳信号。通过 Hilbert 变换得到 $\bar{c}_i(t)$:

$$\bar{c}_i(t) = H[c_i(t)] = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{c_i(\tau)}{t-\tau} d\tau, i=1, \dots, n \quad (2)$$

利用 $c_i(t)$ 和 $\bar{c}_i(t)$ 构造复信号

$$z_i(t) = c_i(t) + j\bar{c}_i(t) = a_i(t) e^{j\theta_i(t)}, i=1, \dots, n \quad (3)$$

从而可以求出 $c_i(t)$ 的瞬时振幅 $a_i(t)$ 、相位 $\theta_i(t)$ 和频率 $\omega_i(t)$ 。

瞬时振幅

$$a_i(t) = \sqrt{c_i^2(t) + \bar{c}_i^2(t)}, i=1, \dots, n \quad (4)$$

瞬时相位

$$\theta_i(t) = \arctan \frac{\bar{c}_i(t)}{c_i(t)}, i=1, \dots, n \quad (5)$$

瞬时频率

$$\omega_i(t) = \frac{d\theta_i(t)}{dt}, i=1, \dots, n \quad (6)$$

3.2 算法并行化分析

并行计算中常见的并行计算模式有数据并行、流水线并行、进程农庄、工作池等^[8]。并行算法设计一般可以分为可行算法的描述与分析、工作分解、选择编程(实现)模型和性能检查及优化 4 个阶段。

3.2.1 EMD 算法

根据经验模态分解方法的过程,给出具体算法^[9,10]:

Step1 初始化: $r_1(t) = x(t), i=1, k=0$ 。

Step2 分解出第 i 阶 IMF 分量 $c_i(t)$:

(1) 设置 $h_1(t) = r_1(t)$;

(2) 查找出 $h_k(t)$ 中所有极大值 MAX_k 和极小值 MIN_k ;

(3) 利用三次样条插值函数对极大值 MAX_k 和极小值 MIN_k 进行拟合,求出上下包络线 $e_+(t)$ 和 $e_-(t)$;

(4) 计算上下包络线均值 $m_+(t)$;

(5) $h_{k+1}(t) = h_k(t) - m_k(t)$;

(6) 计算两次 $h_k(t)$ 序列之间的总方差 $\sigma = \frac{\sum [h_{k+1}(t) - h_k(t)]^2}{\sum [h_k(t)]^2}$ 。若 $\sigma \leq \epsilon$ (ϵ 为筛分门限), 则 $c_i(t) = h_k(t)$; 否则, 令 $k = k+1$, 转(2)迭代。

Step3 $r_{k+1}(t) = r_k(t) - c_{k+1}(t)$, 判断余量 $r_{k+1}(t)$ 是否为单调函数或是常量。如果满足条件, 整个 EMD 分解过程结束。

3.2.2 依赖性分析

为了更好地实现地震瞬时属性提取算法的并行化, 本文从地震数据和算法流程中进行依赖性分析, 将相对独立的运算部分进行并行化设计考虑。

(1) 地震数据分析

地震数据通常以采样点形成数据道集, 通过炮线方向的数据集形成地震剖面, 构造出三维地震的采集数据体(见图 2)。

每条炮线由若干条数据道组成, 每条数据道可以看成为一个时间函数的信号序列 $x(t)$ 。从而采集数据体可以表述为地震数据道按照各道对应的方位进行排列的数据集 $D = \{x_k(t)\}, k=1, \dots, M$, 其中 $x_k(t)$ 为其中一条道线数据, 即一个地震信号序列。 M 为采集数据的总道数。

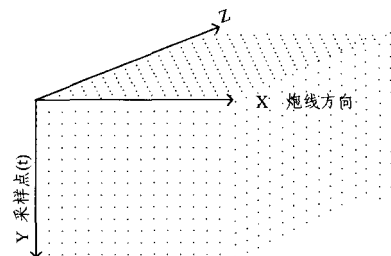


图2 采集数据体结构示意图

在地震资料处理过程中, 各道数据之间(地震信号序列)相对独立, 经验模态分解过程不会出现交叉的情况。因此, 从地震数据的角度分析, 可以考虑实现各条道线数据上的并行化处理。

(2) 算法流程分析

从 EMD 算法流程来看,它是一个双循环迭代的过程,如图 3 所示。内循环①通过迭代分离出一个分量,而外循环②实现符合 IMF 条件的全部分量提取。

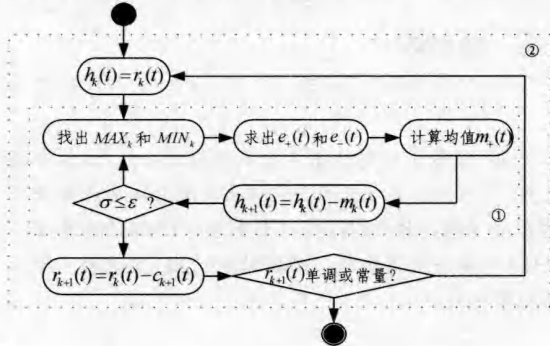


图 3 EMD 双循环迭代

不妨将算法流程看成拓扑结构图,该图中的后活动是在前活动完成的前提下开始的,具有先后次序,没有并行的活动路径。内循环每次迭代开始的输入依赖于上一次迭代的输出,内循环的每一次分量提取迭代的开始依赖于外循环输入,而外循环的输入则与内循环上一次分量迭代输出直接关联。因此,算法无法实现流程上的并行设计。

3.2.3 CUDA 编程模型

CUDA 并行编程模型的核心为线程组层次、共享存储器以及栅栏同步 3 个重要抽象。这些抽象提供了细粒度的数据并行和线程并行性,嵌套在粗粒度数据并行性和任务并行中。可以将问题分解为粗粒度的子问题,这些子问题可以独立并行地解决。再把这些粗粒度的子问题分解成更小的片段,这些片段一起并行处理。每个子任务可以在任何一个可用的处理核心运行^[11]。

在 CUDA 架构下,程序分为 CPU 执行的主机端和 GPU 上执行的设备端。GPU 上程序执行的最小单位是线程(thread),多个线程可以组成块(block)。同一块中的线程共享一块大小有限的共享内存,在同一个流处理器 SM 中被调度执行。多个块组成一个网格(grid),它用来表示一系列可以被并行执行的块集合。运行在 GPU 上的 CUDA 并行计算函数称为内核函数(kernel function)。一个内核函数并不是一个完整的程序,而是整个 CUDA 程序中的一个可以被并行执行的步骤。

3.3 GPU 架构下 EMD 算法的设计

通过对算法的依赖性分析不难明确 GPU 架构下实现 EMD 算法的并行设计思路。以地震数据道集为处理目标,实现多任务的并行处理设计。

地震数据体往往较大,GPU 内存不足以一次性存放所有的数据体。实际处理过程中,每次从主机内存中选择一张地震剖面数据传递到 GPU 全局存储器,剖面数据由一定数量的地震道组成,作为多任务并行处理的数据输入。

每个子任务负责处理一条道线数据,对应 GPU 上调度运行的一个线程。由于 GPU 每个流多处理器 SM 可以同时调度的线程数有限,每个线程块中的所有线程将在同一个 SM 中被调度运行直至完成。因此,需要将所有子任务划分成多个任务块,以充分利用 GPU 的流处理器实现并行处理。根据地震数据道集的数量,将道集分成 $n \times x$ 矩阵网格。负责处理 n 个道集的子任务线程组成一个线程块,共有 x 个线程

块(最后一块的线程数可能小于 n)。每个子任务线程块每次被调度 m 个线程投入运行,线程块之间互不影响,如图 4 所示。

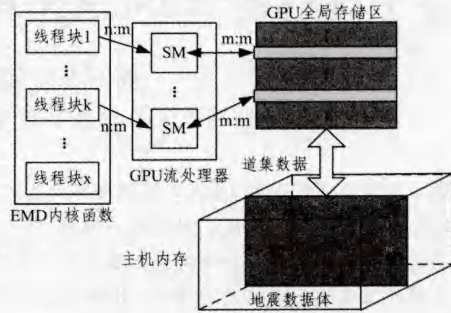


图 4 EMD 并行处理过程

被调度运行的各子任务线程将根据其线程序号去 GPU 全局存储区查找对应的道线数据进行 EMD 处理,并将处理完成后的结果输出到对应的原始道线数据区。为了节省 GPU 存储空间,每条道线处理完成的输出数据将覆盖原来的道线数据空间,所有道线处理完成后,其空间将被处理后的输出结果完全取代。

GPU 全局存储区中的剖面数据完成处理后,将反馈传递给主机内存,然后再重新选择一张未被处理的地震剖面数据传递给 GPU 全局存储区重复上面的过程,直到整个地震数据体处理全部完成。

4 实验结果

本文实验在 NVIDIA Quadro 5000 GPU 和 Intel Xeon X5670 2.93GHz 双核 CPU 配置环境下进行。以不同地震道集数据体为测试目标,提取地震瞬时属性的 IMF 分量。图 5 为不同地震采样分辨率下 GPU 与 CPU 不同架构下 EMD 算法的运行时间对比。从对比图上可以看出 GPU 架构下提取地震瞬时参数的算法运行效率较 CPU 具有明显优势。在测试数据中,道数与采样点为 1501 和 6000 时,GPU 加速比(见图 6)达到了 8.66 倍。

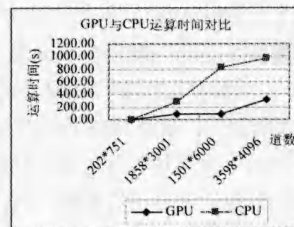


图 5 GPU 与 CPU 运算时间对比

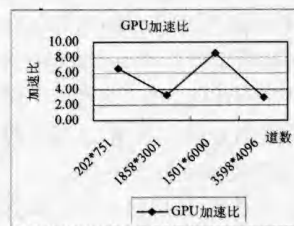


图 6 GPU 加速比

运行于 GPU 的内核函数基于道线进行线程调度并行处理,CPU 下该算法则是基于循环迭代实现。对实验数据进一

(下转第 414 页)

(2)利用加权与未加权 DBSCAN 算法对农安县农安镇 2011 年土壤中碱解 N、有效 P 和速效 K 3 种土壤养分数据进行聚类的对比分析,结果表明加权 DBSCAN 算法比未加权的算法易于选取 Eps 和 minPts 两个参数,改进后的 DBSCAN 算法聚类效果更好。

虽然 DBSCAN 算法能够发现任意形状的簇,但土壤肥力数据具有空间性、不确定性、复杂性、地域性等特点,当数据量较大时,所要求的内存支持大;当数据分布不均匀时,由于使用统一的全局变量,使得聚类的效果差。另外,DBSCAN 算法需要人为确定 Eps 和 minPts 2 个参数,导致聚类过程需人工干预才能进行。本文提出的加权 DBSCAN 算法未能综合考虑上述问题,证实该算法对海量数据集聚类的有效性,如何简化聚类算法并结合多年多个乡(镇)以及土壤类型的数据进行验证,还需要进一步的深入研究。

参 考 文 献

[1] 卜东波. 聚类/分类理论研究及其在文本挖掘中的应用[D]. 北京:中国科学院技术研究所,2000

[2] Ester, Martin, Kriegl H P, et al. A Density Based Algorithm for Discovering Clusters in Large Spatial Data bases with Noise [C]// Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96). Portland, Oregon, 1996

[3] Li Liang-hou, Li Ji-yue. Application of Clustering Analysis in

classifying Site Type and Evaluating Soil Fertility[C]// 2010 Third International Conference on Education Technology and Training (ETT 2010). 2010:468-471

[4] Turner B L, Meyer W B. Land use and land cover in global environmental change: considerations for study [J]. Int. Soil Sci. J., 1991, 130:669-680

[5] 张书慧,马成林,李伟,等. 变量施肥对玉米产量及土壤养分影响的试验[J]. 农业工程学报,2006,22(8):64-67

[6] 陈桂芬,曹丽英,王国伟. 加权空间模糊动态聚类算法在土壤肥力评价中的应用[J]. 中国农业科学,2009,42(10):3559-3563

[7] 周水庚,等. 基于数据分区的 DBSCAN 算法[J]. 计算机研究与发展,2000,37(10)

[8] 盛骤,谢式千,潘承毅. 概率论与数理统计(第3版)[M]. 北京:高等教育出版社,2004

[9] Umeda M, Kaho T, Iida M, et al. Effect of variable rate fertilizing for paddy field[C]//2001 ASAE annual international meeting, 2001, Number. 01(Part. II)

[10] Wittry D J, Mallarino A P. Comparison of uniform and variable rate phosphorus fertilization for corn-soybean rotations[J]. Agronomy Journal, 2004, 96(1):26-33

[11] 任兴平,何忠龙,孟增辉. 改进 DBSCAN 算法中参数 Eps 值的确定[J]. 现代化技术,2007,(11):120-121

[12] 刘志勇,耿新青. 基于模糊聚类的文本挖掘算法[J]. 计算机工程,2009,35(5):44-45

(上接第 411 页)

步分析发现,GPU 架构下的 EMD 算法对道线数更敏感,而 CPU 架构下的 EMD 算法则似乎对采样点数更敏感些。

图 7 给出了原始地震信号剖面图与 EMD 分解后利用 Hilbert 变换^[12]提取的瞬时频率 IMF 第 1 和第 2 分量。从 IMF 提取的分量图上看,高频分量包含了较多的噪音,而低频分量所刻画的能量强弱表现较高频分量要更清晰、更丰富。

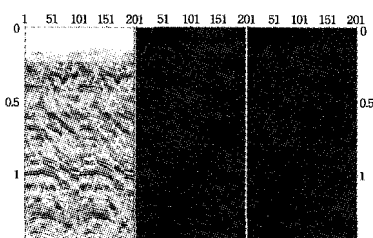


图 7 原始信号与瞬时频率第 1 和第 2 分量

参 考 文 献

[1] 皮红梅,刘财,王典. 利用 Hilbert-Huang 变换提取地震信号瞬时参数[J]. 石油地球物理勘探,2007,42(4)

[2] Xie K, Wu P, Yang S. GPU and CPU cooperation parallel visualization for large seismic data[J]. Electronics Letters, 2010(17)

[3] Waskito P, Miwa S, Mitsukura Y, et al. Parallelizing Hilbert-Huang transform on a GPU[J]. Networking and Computing

(ICNC), 2010

[4] Yu Wen-mao, Xie Kai, Yu Huo-quan, et al. Hilbert-Huang Transformation of Large Seismic Data Based on GPU[C]// Intelligence Science and Information Engineering (ISIE), 2011

[5] NVIDIA 官方网站[OL]. http://www.nvidia.cn/object/product_tesla_C2050_C2070_cn.html

[6] 韩俊刚,刘有耀,张晓. 图形处理器的历史现状和发展趋势[J]. 西安邮电学院学报,2011,16(3)

[7] Huang N E. The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-stationary Time Series Analysis[J]. J. Proc. R. Soc. Lond. A, 1998, 454:903-995

[8] 万剑怡,孙永强,薛锦云. 一种基于设计模式的三阶段并行程序设计方法[J]. 计算机研究与发展,2002,39(3)

[9] Huang N E, Wu Z. A review on Hilbert-Huang transform: method and its applications to geophysical studies[J]. Reviews of Geophysics, 2008, 46(2)

[10] Flandrin P, Rilling G, Goncalves P. Empirical mode decomposition as a filterbank[J]. IEEE SIGNAL Proc Lett., 2004, 11: 112-114

[11] Nvidia Corporation. NVIDIA CUDA C Programming Guide Version 4. 2[Z]. 2012

[12] 刘慧婷,程家兴,张旻. 利用 Hilbert 变换提取信号瞬时特征的算法实现[J]. 微机发展,2003,13(6)