

doi:10.11720/wtyht.2015.5.32

李逢春,王润秋,赵薇薇,等.地震三维观测系统共中心点面元属性的并行计算方法[J].物探与化探,2015,39(5):1074-1078.http://doi.org/10.11720/wtyht.2015.5.32

Li F C, Wang R Q, Zhao W W, et al. Common mid-point bin attribute analysis of three-dimensional seismic geometry using parallel computation method[J]. Geophysical and Geochemical Exploration, 2015, 39(5): 1074-1078. http://doi.org/10.11720/wtyht.2015.5.32

# 地震三维观测系统共中心点面元属性的并行计算方法

李逢春<sup>1,2</sup>, 王润秋<sup>1</sup>, 赵薇薇<sup>2</sup>, 杜清波<sup>2</sup>, 郭武<sup>2</sup>, 王汉钧<sup>2</sup>

(1. 中国石油大学(北京)地球物理与信息工程学院, 北京 102249; 2. 东方地球物理公司采集技术支持部, 河北涿州 072751)

**摘要:** 针对高密度地震勘探下的三维观测系统海量面元属性计算需求, 基于 OpenCL 并行编程模型, 研究并实现了适用于异构环境下面元属性的并行计算方法。根据面元分析算法特点将整个网格分成多个子区域, 子区域可由不同设备以粗粒度方式并行, 而子区域内部以炮检对方式细粒度并行。采用 HP Z820 工作站, 在 CPU+GPU 异构混合平台下的试算结果表明, 异构多核处理速度是 CPU (单核) 的 30 倍以上, 数据生成速率高于 300 MB/s。

**关键词:** 地震观测系统; 面元分析; 并行计算; 异构计算; OpenCL

**中图分类号:** P631.4      **文献标识码:** A      **文章编号:** 1000-8918(2015)05-1074-05

为了合理设计野外观测系统方案, 需要对不同观测系统的共中心点 (common mid-point, CMP) 面元属性进行对比分析。常规 CMP 面元属性分析有覆盖次数、炮检距和炮检方位角。对覆盖次数、炮检距和炮检方位角的直观和精确描述, 是优化观测系统设计的关键因素之一, 而 CMP 面元属性计算则是实现观测系统面元属性分析的基本技术手段<sup>[1-3]</sup>。

近年来, 随着地球物理探测技术的发展, 特别是万道以上地震仪的推广应用, 高密度地震勘探技术正逐渐成为地震勘探技术发展的主流<sup>[4]</sup>。高密度地震勘探通过减小面元尺度和提高空间采样率来增加采样密度。目前, 常规地震勘探面元一般为 25 m × 25 m, 而为了适应未来的石油勘探开发的精度要求, 可能采用 5 m 的空间采样密度, 这对于山地勘探和小断块、薄储层、小砂体、小尺度孔洞的识别以及精细油藏描述具有重要意义<sup>[5]</sup>。高密度地震勘探具有小道距、高覆盖的特点, 因此极大地增加了地震野外采集的数据量。野外生产的接收道数不断增多, 百万乃至千万级炮检点的布设越来越普遍。日益增长的地震采集应用对三维地震观测系统面元属性快速分析和变观提出了越来越高的要求<sup>[6-7]</sup>。由于一个面元的面元属性值是对该面元上覆盖的多

个关系片计算后的统计和累加结果, CMP 面元属性计算的工作量大、耗时长, 对存储容量和 I/O 的要求更高, 成为生产应用的瓶颈。目前, 普遍应用于野外施工设计的软件主要有美国 ION 公司的 Green Mountain 软件<sup>[8]</sup> 和 GEDCO 公司的 OMNI 软件<sup>[9]</sup>, 均面临计算时间长、结果数据量太大导致计算失败等问题。这些软件面对超大观测系统和高覆盖次数 (千次以上) 面元分析任务的挑战, 也显得无能为力。在对 CMP 面元属性分析的过程中, 由于传统的面元属性计算方法是串行执行的, 即依次计算每个面元的面元属性, 因此, 在多核硬件环境下, 不能充分利用计算资源; 由于串行执行 CMP 面元属性计算的效率低, 而不能满足高密度地震勘探带来海量 CMP 面元属性计算的需求, 从而导致了分析 CMP 面元属性时, 效率低、资源利用率低的问题。

图形处理器 GPU 有着高速浮点性能和超高的内存带宽的独特优势, 近年来在地球物理领域得到了广泛而成功的应用。地震勘探目前对高性能计算的直接需求是叠前偏移、叠前反演和全波形反演<sup>[5]</sup>。张兵等<sup>[10]</sup> 实现了裂步法二维地震叠前深度偏移处理的 GPU 计算。刘国峰等<sup>[11]</sup> 基于 CUDA 编程环境实现了台式 GPU/CPU 协同并行计算, 并成

功应用到地震偏移成像。张慧等<sup>[12]</sup>采用震源重构方法解决了逆时偏移互相关成像条件需要的海量存储问题,引入 GPU 解决逆时偏移计算效率低的问题。张猛等<sup>[13]</sup>研究提出基于 CPU/GPU 异构平台的时空域声波方程全波形反演算法实现流程。

开放计算语言(open computing language, OpenCL)作为面向异构系统通用目的并行编程行业标准,得到包括 Intel、NVIDIA 和 AMD 等主流芯片厂商的支持<sup>[14-15]</sup>。OpenCL 为软件开发人员提供了统一的异构系统的并行编程环境,屏蔽了基于 SDK 编程的很多复杂硬件细节,提高了并行编程的生产率。针对 CMP 面元分析特点,笔者采用基于 OpenCL 的面元网格分区并行计算方法。并行设计中将计算任务沿纵横向分配成多个子区域,每个子区域由 CPU/GPU 设备进行并行计算,最后通过 10 万道接收观测系统验证文中并行计算方法的效率。

## 1 CMP 面元属性计算原理

按地质任务建立满覆盖观测系统,利用已知观测系统关系片得到炮检中心点和偏移距,并根据炮检中心点得到面元覆盖次数。常规面元属性计算原理如图 1 所示,面元宽为  $w$ ,高为  $h$ ,面元总行数  $N_{row}$ ,面元总列数  $N_{col}$ ,面元线方位角为  $\theta$ ,炮点到检波点的距离  $d_{SR}$  为炮检偏移距。利用三维观测系统炮检关系数据,得到炮检对中心点  $M$  的大地坐标  $M(X, Y)$ 。根据面元划分参数将大地坐标  $M(X, Y)$  转换为相对坐标  $M(x, y)$ ,中心点  $M(x, y)$  在相对坐标系下的坐标为

$$\begin{cases} x = (Y - O_x) \times \sin\theta + (X - O_y) \times \cos\theta, \\ y = -(Y - O_x) \times \cos\theta + (X - O_y) \times \sin\theta. \end{cases} \quad (1)$$

式中: $O_x, O_y$  为观测系统大地坐标原点。

根据中心点  $M(x, y)$  相对坐标确定所在面元行列位置分别为  $i=y/h, j=x/w$ 。统计出一个面元内累计落入的炮检对中心点个数即为该面元的覆盖次数。

炮检距也称偏移距,是具有炮检排列关系的炮点与检波点的距离。方位角是炮检矢量与正北方向顺时针旋转所形成的夹角。可以采用公式

$$D = \sqrt{(s_x - r_x)^2 + (s_y - r_y)^2}, \quad (2)$$

$$\beta = \arctan[(s_x - r_x)/(s_y - r_y)]. \quad (3)$$

计算得到炮检距和炮检方位角。其中: $D$  为炮检距, $\beta$  为炮检方位角, $(s_x, s_y)$  为炮点在大地坐标系下的坐标值, $(r_x, r_y)$  为检波点在大地坐标系下的坐标值。

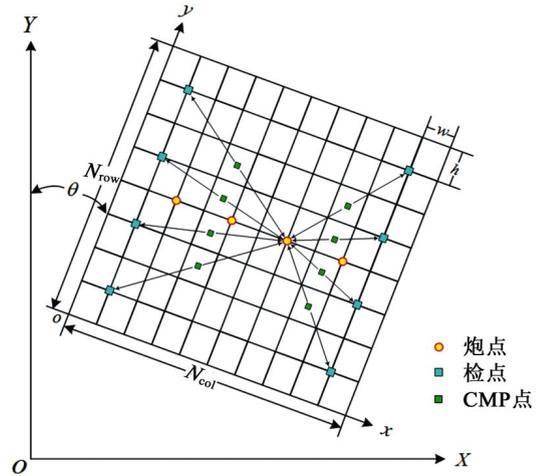


图 1 面元属性计算原理示意

## 2 基于 OpenCL 的 CMP 面元并行计算设计

CMP 面元属性分析的主要计算负荷在于大量的炮检对循环,任何网格属性值计算只需要周围有限个炮检对即可,因此计算过程具有良好的局部性,非常有利于并行计算。首先,读取观测系统数据并将面元网格分成多个子区域(分区);其次,在 CPU 端创建 OpenCL 内核执行所需的运行环境和资源;最后在多设备上执行并行面元计算任务。

算法的核心部分有两点:一是如何根据计算设备的缓存大小将面元网格分区;二是子区域内部炮检对的并行方法。

### 2.1 面元网格分区并行

炮点激发,由具有炮检关系的全部接收点接收,则同一关系片内所有 CMP 点必定落在关系片所在的包围矩形区域内。据此可按关系片包围矩形与分区的相对空间位置将全部关系片映射到各个分区,处于分区边界的关系片可能同时属于几个相邻分区。计算分区作为多设备粗粒度并行基本单位,由每个设备(CPU/GPU)独立负责计算,计算结果拷贝到设备全局内存。由于各分区炮检对 CMP 点叠加过程是独立的,分区间不存在“写后读”、“读后写”、“写后写”的数据竞争情况。

如图 2 所示,针对 CMP 面元计算特点,将三维观测系统待分析区域沿纵横向划分为多个计算分区( $m \times n$  面元格网)。假设每个 CMP 点占用 16 个字节,则分区计算过程中每个网格需要申请的缓存为  $F_{max} \times 16$  个字节。分区大小由设备端开辟的属性数据缓存大小决定,分区的数量由纵横向面元网格数决定。每个分区网格数应满足

$$C = \lfloor F_{max} \times \varphi \times m \times n \rfloor \quad (4)$$

的数量关系。其中: $C$  为设备端可分配的缓存大小

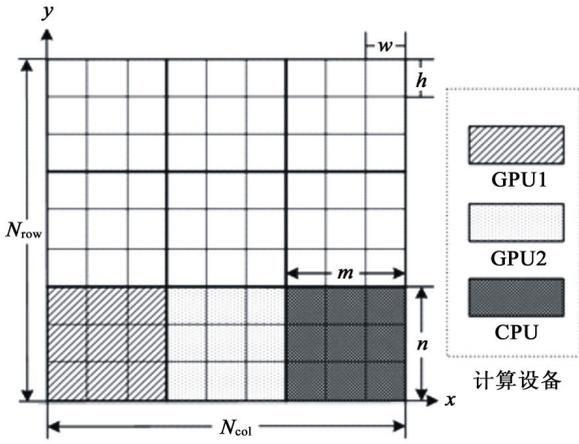


图2 面元网格分区并行示意

(单位:字节),  $\varphi$  为 CMP 点存储字节数,  $F_{max}$  为最大覆盖次数,  $\lfloor \cdot \rfloor$  为向下取整符号。

### 2.2 分区内炮检对并行

本文算法提速的关键正是利用了异构设备大量的线程级并行,将面元属性分析的主要计算量转移至异构多设备端执行,尤其是在超大观测系统面元计算时获得了理想的加速比。对具体的设备而言,要提高算法并行度,就是要使设备运算单元尽可能均匀地分配到计算任务,这是提高计算效率的重要举措。笔者利用 OpenCL 内核(kernel)计算关系片内部每对炮检对的属性值,分区内所包含的若干个关系片按照串行顺序执行。根据炮检关系建立炮检对与 OpenCL 执行模型的映射关系。如图 3 所示,设有  $m$  炮  $n$  道关系片,炮点集合  $S = \{S_0, S_1, \dots, S_{m-1}\}$ ,接收点集合  $R = \{R_0, R_1, \dots, R_{n-1}\}$ ,则炮检对集合为

$$SR = \{S_0R_0, S_0R_1, \dots, S_{m-1}R_{n-2}, S_{m-1}R_{n-1}\}, \quad (5)$$

其中,集合  $SR$  中炮检对个数为  $m \times n$ 。

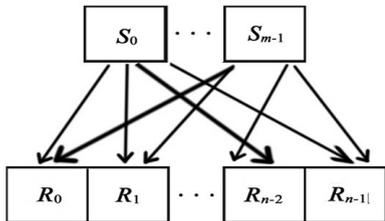


图3 关系片内炮检关系示意

每一个实际执行计算的线程称为一个工作项(work-item),负责一个炮检对的属性值计算。如图 4 所示,炮检对集合  $SR$  可映射为一个一维的全局工作项索引空间,工作项进一步组织成工作组(work-group),则工作组分组数和全局工作项数目分别为

$$N_{GROUPSNUM} = \lceil (m \times n) / \min(L_0, G_{max}) \rceil, \quad (6)$$

$N_{GLOBALSIZE} = N_{GROUPSNUM} \times \min(L_0, G_{max})$ 。 (7)  
其中: $N_{GROUPSNUM}$  为工作项分组数目,  $m$  为炮点数,  $n$  为接收点数,  $L_0$  为设备工作项第一维尺寸,  $G_{max}$  为设备硬件所允许的工作组最大尺寸,  $\lceil \cdot \rceil$  为向上取整符号,  $N_{GLOBALSIZE}$  为全局工作项数目。

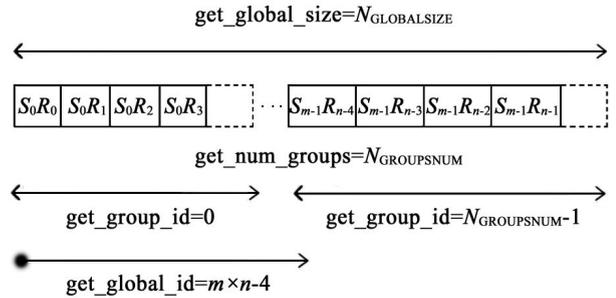


图4 炮检对映射到工作项索引空间示意

每个工作组对应  $\min(L_0, G_{max})$  个炮检对,而每个工作项对应其中一个炮检对。激发点集合  $S$  与接收点集合  $R$  作为输入数据存放于设备全局内存。在工作项内部,基于 OpenCL 技术在设备上进行的属性计算过程可分解成:根据工作项全局标示符从全局内存拷贝激发点和接收点数据到本地内存,计算中心点坐标、偏移距和方位角,将计算结果数据拷贝回全局缓存。

### 2.3 并行面元属性计算实现流程

结合 CMP 面元分析算法原理及 GPU 等多核处理器架构特点,设计了异构平台下基于 OpenCL 的并行计算流程,如图 5 所示。

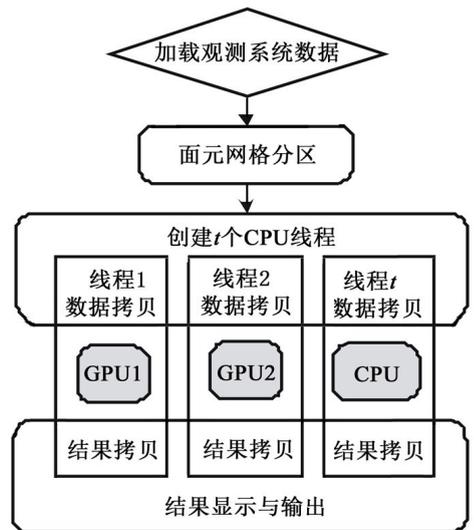


图5 异构并行面元属性计算流程

步骤一:读取观测系统数据并将面元网格分成多个子区域(分区)。

读取面元网格划分参数和设备的执行配置参

数,根据计算设备可用内存大小,由式(4)得到面元网格分区大小和分区个数。计算每个关系片包络矩形,得到每个分区所包含的关系片集合,然后将每个分区作为基本计算任务放入任务队列 TaskQueue。

步骤二:在 CPU 端创建 OpenCL 内核执行所需的运行环境和资源。

为每个计算设备创建 OpenCL 缓存对象及命令队列;加载并编译内核代码,创建程序执行体。为每个设备创建一个执行任务的 CPU 线程。

步骤三:在多设备上执行并行面元计算任务。

以步骤一得到的分区及其所属关系片集合作为基本的计算任务,运行在各计算设备上的任务执行线程向任务队列 TaskQueue 申请计算任务。任务队列 TaskQueue 充当向各任务执行线程分发计算任务的角色。调用 OpenCL 内核 Kernel 计算不同炮检对的中心点、偏移距和方位角。将计算结果从设备端缓冲对象读入到主机内存,并跳转到步骤三。重复步骤三直到任务队列 TaskQueue 为空,则算法结束。

### 3 实验结果及其分析

#### 3.1 实验数据与环境

以 L120S8T115 200(120 线 8 炮 115 200 道,960 点/接收线,接收点距 12.5 m,接收线距为 100 m,炮点距 12.5 m,炮线距 100 m、滚动距为 100 m)的正交式观测系统为例,纵向(inline)和横向(xline)方向各滚动 200 次,得到勘探区域 inline 方向长 31 887.5 m,xline 方向宽 318 000 m。总激发点数 320 000,总接收点数 814 088,关系片个数 40 000。面元大小为 6.25 m×6.25 m,则 inline 方向面元网格数 5 102,xline 方向面元网格数 5 088。满覆盖次数 3 600 次,面元属性计算结果数据总量达到 549 GB。

本实验采用惠普新一代 Z 系列图形工作站 HP Z820,CPU 采用 12 核英特尔至强 E5-2600 系列处理器,支持双处理器。GPU 分别为 NVIDIA Quadro K5000/4GB 和 NVIDIA Tesla K40c/12GB。操作系统为 win7 专业版 64 位,驱动支持 OpenCL1.2。具体参数如表 1 所示。

表 1 实验环境

设备	参数
CPU	Intel Xeon E5-2697v2,2.7 GHz,128 GB Memory 24Cores/48Threads
GPU	NVIDIA Quadro K5000,4 GB Memory NVIDIA Tesla K40c,12GB Memory

#### 3.2 计算效率分析

高密度超大观测系统面元属性计算难以在生产中推广应用的一个重要原因是耗时长、微机磁盘占

用量太大。共中心点面元分析算法效率依赖分区的划分方式,需要综合平衡 GPU 显存和计算子区域的大小。文中选取 3 种不同的分区大小进行计算,即 100 面元×100 面元(549 MB)、200 面元×200 面元(2 197 MB)、250 面元×250 面元(3 433 MB)。分别采用 4 种并行加速策略,每种加速策略按分区从小到大的顺序计算得到加速比如下。

策略 1:在 Intel CPU 平台上测得加速比分别为 10.4、13.2、8.2;属性数据的生成速率分别为 100、278.6、358.8 MB/s。

策略 2:在 NVIDIA Quadro K5000 平台上测试加速比分别为 12.8、14.2、11.6;属性数据的生成速率分别为 123.6、300.1、506.9 MB/s。

策略 3:在 NVIDIA Tesla K40c 平台上的测试加速比分别为 16.5、16.9、13.9;属性数据的生成速率分别为 158.5、356.5、606.4 MB/s。

策略 4:在 CPU+GPU 混合平台上测试加速比分别为 38.1、43.6、29.6;属性数据的生成速率分别为 366.7、918.6、1 286.4 MB/s。

从并行计算加速比的对比中不难看出,在各个单一的硬件平台上所获得加速比普遍在 10 倍以上,而在 CPU+GPU 混合平台上取得的加速比达到 30~40 倍。从表 2 中可以看出,随着分区增大,即开辟较大的缓存,计算效率呈上升趋势。这是因为计算子分区越大,处于相邻分区的重叠关系片数量越少,从而减少了冗余计算量。另外,从属性数据的生成速率来看,上述 4 种策略的数据生成速率普遍高于目前主流机械磁盘 I/O 性能。

表 2 CPU 及 CPU+GPU 计算时间比较

CPU 串行/s	CPU 并行/s	k5000/s	tesla k40c/s	CPU+k5000+ tesla k40c/s	缓存/MB
58364	5601	4549	3547	1533	549
26689	2018	1873	1577	612	2197
12914	1567	1109	927	437	3433

### 4 结论

笔者针对高密度地震勘探下的三维观测系统海量面元属性计算性能问题,研究了 CPU/GPU 环境下协同并行的高效算法。试算结果表明本算法对接收道数量大、布设密度高、勘探面积大的三维观测系统,可以有效提高面元属性计算效率。本算法一方面充分利用微机多核资源,另一方面以实时计算得到属性数据的方式替代原有先计算再存盘的方式,极大地节省了微机磁盘空间,更方便实际生产应用。

对于本文共中心点面元属性并行算法的进一步研究,可以从以下几方面考虑:①目前的计算任务划

分与设备缓存大小相关,且按相同分区大小进行划分,导致具有较大内存的设备缓存不能充分利用,如何进一步合理划分任务还有待深入的研究和实验;  
②对多个计算设备而言,设备间的负载均衡问题需要在后续的研究中加以考虑。

### 参考文献:

- [1] 李培明,何永清.三维地震勘探设计[M].北京:石油工业出版社,2008.
- [2] 尹成,吕公河,田继东,等.三维观测系统属性分析与优化设计[J].石油地球物理勘探,2005,40(5):495-498.
- [3] 黄迎雪,代金芝,黄鹏.三维观测系统属性分析研究[J].工程地球物理学报,2012,9(1):25-28.
- [4] 刘振武,撒利明,董世泰,等.中国石油高密度地震技术的实践与未来[J].石油勘探与开发,2009,36(2):129-135.
- [5] 张军华,臧胜涛,单联瑜,等.高性能计算的发展现状及趋势[J].石油地球物理勘探,2010,45(6):918-925.
- [6] 刘振武,撒利明,董世泰,等.中国石油物探技术现状及发展方向[J].石油勘探与开发,2010,37(1):1-10.
- [7] 赵改善.地球物理软件技术发展趋势与战略研究[J].勘探地球

物理进展,2010,33(2):77-86.

- [8] ION Corporation. Acquisition survey design and 3D modeling software[2015-03-12]. [http://www.iongeo.com/content/documents/Resource%20Center/Brochures%20and%20Data%20Sheets/Data%20Sheets/Concept\\_Systems/DS\\_CSL\\_MESA\\_14\\_Features.pdf](http://www.iongeo.com/content/documents/Resource%20Center/Brochures%20and%20Data%20Sheets/Data%20Sheets/Concept_Systems/DS_CSL_MESA_14_Features.pdf)
- [9] GEDCO Corporation. 3D seismic design using OMNI 3D[2015-03-12]. [http://www.gedco.com/Uploads/Site\\_Files\\_OLD/OMNI3D\\_Workshop.pdf](http://www.gedco.com/Uploads/Site_Files_OLD/OMNI3D_Workshop.pdf)
- [10] 张兵,赵改善,黄骏,等.地震叠前深度偏移在 CUDA 平台上的实现[J].勘探地球物理进展,2008,31(6):427-432.
- [11] 刘国峰,刘钦,李博,等.油气勘探地震资料处理 GPU/CPU 协同并行计算[J].2009,24(5):1671-1678.
- [12] 张慧,蔡其新,秦广胜,等.基于 GPU 并行加速的逆时偏移成像方法[J].石油地球物理勘探,2013,48(5):707-710.
- [13] 张猛,王华忠,任浩然,等.基于 CPU/GPU 异构平台的全波形反演及其实用化分析[J].石油物探,2014,53(4):461-467.
- [14] Khronos OpenCL Working Group. The OpenCL Specification[S]. Version 1.2, 2011.
- [15] 苏金国,李璜,杨健康,译. OpenCL 编程指南[M].北京:机械工业出版社,2013.

## Common mid-point bin attribute analysis of three-dimensional seismic geometry using parallel computation method

LI Feng-Chun<sup>1,2</sup>, WANG Run-Qiu<sup>1</sup>, ZHAO Wei-Wei<sup>2</sup>, DU Qing-Bo<sup>2</sup>, GUO Wu<sup>2</sup>, WANG Han-Jun<sup>2</sup>

(1. College of Geophysics and Information Engineering, China University of Petroleum, Beijing 102249, China; 2. Acquisition Technique Supports, BGP Inc., CNPC, Zhuozhou 072751, China)

**Abstract:** For the purpose of common mid-point bin attribute analysis from high density exploration and three-dimensional seismic geometry, this paper analyzes the calculation characteristics of bin analysis, and proposes an efficient parallel algorithm for bin attribute analysis based on Open Computing Language (OpenCL) heterogeneous programming model. The algorithm divides bin grid into several sub-regions under hardware memory constraint. Tasks of sub-regions are executed with coarse-grained parallelization by assigning task to different devices, whereas pairs of source-receiver in sub-region are executed with fine-grained parallelization. With HP's Z820 desktop workstation, experiment was carried out with high density three-dimensional geometry from seismic exploration. The experimental results show that the parallel algorithm takes full advantage of multiprocessor's parallel processing capability, and the processing speed is about 30 times faster than that on CPU-based implementation (single core) under the condition that the precision is not changed.

**Key words:** seismic geometry; bin analysis; parallel computing; heterogeneous computing; OpenCL

**作者简介:** 李逢春(1976-),男,重庆市人,2006年6月在武汉大学获得工学博士学位,主要研究方向为并行计算、三维可视化及石油勘探软件开发。E-mail: lifcmail@sina.com