基于多线程的地震相干体属性提取算法①

杨尚琴, 许自龙, 洪承煜

(中国石油化工股份有限公司 石油物探技术研究院, 南京 211103)

摘 要: 为了充分发挥计算机的多核优势, 提高地震数据相干体的计算速度, 通过研究在多核上的多线程并行技术, 完成了并行相干体算法的设计与实现, 并分别对串行和并行算法进行性能比较测试. 测试结果表明: Pthread 多线程技术可以充分利用多核资源, 取得比较理想的线性加速比, 且提高了系统的计算效率, 非常适合于大数据量的地震数据处理的应用.

关键词: 相干体; 地震数据处理; Pthread; 多线程技术; 并行计算

Attribute Extraction Algorithm of Seismic Coherence Based on Multi-Threading

YANG Shang-Qin, XU Zi-Long, HONG Cheng-Yu

(SINOPEC Geophysical Research Institute, Nanjing 211103, China)

Abstract: For taking full advantage of multi-core computers and improving computing speed of coherent cube of seismic data, a coherence parallel algorithm with the research of multi-core parallel technology has been designed and implemented. The comparison test of the performance between the serial and the parallel algorithm has been completed. The result shows that using Pthread can make full use of multi-core resources and achieve an ideal linear speedup. The result also indicates that using Pthread can improve the computational efficiency of the system and be very suitable for processing large-scale seismic data.

Key words: coherence; seismic data processing; Pthread; multi-threading; parallel computation

随着物探技术的发展, 地震属性技术在地震资料解释中的作用越来越大, 例如在储层预测和表征方面, 利用地震属性技术可以预测岩性和有利储集体, 描述油藏特征及孔隙度变化, 寻找死油区, 甚至于监测流体运动和进行其他综合研究. 而地震数据相干体属性是众多地震属性中相当重要又常用的一种, 它是通过对三维数据体的不连续性进行分析, 可识别构造和断层的分布, 使解释人员在解释之前就能获得研究区域粗略的构造几何形态和断层分布情况, 避免解释的随意[1,2].

但是, 相干体的提取算法及流程比较复杂, 尤其面对现今规模愈发庞大的三维地震数据体, 计算过程耗费的时间也越来越长, 这已经不能满足目前科研工作周期越来越短的要求. 因此为了缩短工作周期达到提高工作效率的目的, 需要寻找当前最有效的并行技术, 并根据现有硬件的特性, 充分利用多核计算机来帮助我们完成工作. 目前, 可以使用的并行编程技术主要包括针对 CPU 的 OpenMP、Pthread 等和针对 GPU 的 CUDA 技术. (1)CUDA 并行利用的 GPU 通常在显卡上, 需要 NVIDIA 等专门的显卡支持, 代价比较高, 且目前 CUDA 并行计算的数据要经过内存才能加载到 GPU, 考虑到本算法的特定性——没有大规模矩阵数值运算, 不能很好的解决 I/O 与计算并行问题, 本文不讨论此方案. (2)OpenMP 通过对原有的串行代码插入一些指导性的注释, 并进行必要的修改, 可以快速的实现并行编程. 但这些 OpenMP 指导语句需要高版本编译器的支持[3], 本文在此也不讨论此方案. 本论文将使用足够轻量级且有很好跨平台能力的 Pthread 编程来实现该系统中包含的地震相干体属性提取算法的多线程并行, 以提高解释系统的运行速度、能效比及

① 收稿时间:2012-04-14;收到修改稿时间:2012-05-16

多核 CPU 的利用率. Pthread 即 POSIX Thread, 是基于 POSIX 标准的多线程 API, 它的使用灵活, 且合理的应用可以达到高效的并行, 但同时它的灵活性也给编程带来了复杂性[4,5], 所以本文的实现是基于在自己实现的线程池上进行的算法并行化, 这既降低了编程的难度, 也提高了今后代码的可移植性.

本文重点研究相干体的并行算法, 并对算法进行并行化的实现. 首先由提取相干体属性的串行算法引出基于 Pthread 多线程库的并行算法, 然后对该并行相干体算法进行不同规模及不同影响参数下的性能测试.

1 相干算法现状

三维相干体技术是 20 世纪 90 年代后期兴起的一项十分有效的地震解释技术, 它利用三维地震数据体中相邻道之间地震信号的相似性, 来描述地层和岩性的横向非均匀性.

在实际地震勘探资料解释中, 利用特征值分析方法进行相干分析和处理的方法有:

1)基于互相关的 C1 相干算法; (2)基于相似性的 C2 相干算法[6]; (3)于数据协方差矩阵本征结构的 C3 相干算法. 其中 C3 相干算法从理论上讲优于 C1 相干算法和 C2 相干算法, 下面我们先简单介绍基于数据协方差矩阵本征结构的 C3 相干算法[7,8].

假设在地震三维偏移体中, 截取相邻 J 道 N 个样点组成一个 $N \bullet J$ 的地震子体构成矩阵 D:

$$D = \left[d_{nj} \right]_{N \times J} \tag{1}$$

D 中每列代表一个有 N 个样点的地震道(第 j 道), 每行为 J 道中同一个时间样点(第 n 个样点), d_{nj} 即为每 j 道的第 n 个样点. J 维变量的正交关系在数学上可用协方差矩阵来表示, 则 D 的协方差矩阵 C 可用下述公式来表示:

$$C_{J \times J} = D^T{}_{J \times N} \bullet D_{N \times J} = \sum_{n=1}^{N} d_n d_n^T \tag{2}$$

定义 $Tr(C)$ 为:

$$Tr(C) = \sum_{j=1}^{J} \lambda_j = \sum_{j=1}^{J} C_{jj} = \sum_{j=1}^{J} \sum_{n=1}^{N} d_{nj}^2 \tag{3}$$

式中 $Tr(C)$ 为协方差矩阵的迹, 代表分析窗口中地震数据的能量和, 也等于协方差矩阵 C 特征值之和; 而 λ_j 为 C 的本征值. 而本算法最终要求的是最大相干值, 即:

$$E_C = \frac{\max(\lambda_j)}{\sum \lambda_j} = \frac{\max(\lambda_j)}{Tr(C)} = \frac{\max(\lambda_j)}{\sum_{j=1}^{N} C_{jj}} \tag{4}$$

由于, 地震数据规模日益庞大, 而地震解释项目周期越来越短, 串行的 C3 相干体算法的计算速度已经不能满足当前的计算需求. 因此亟需在多核 CPU 上利用多线程并行技术来充分发挥多核的优势, 从而提高相干体的计算速度, 缩短项目周期, 进而节约成本.

2 基于多线程库的相干并行算法

由于 C3 相干算法在分辨率和信噪比方面最优, 但是计算成本最高, 需充分的分解任务、数据及数据流的相关性. 通过算法的计算热点测试, 得协方差矩阵为计算热点. 从算法功能分析得, 它可通过多个线程对不同的数据对象执行相同的操作, 即按线分解一定范围的地震数据来实现; 从算法内部实现分析得, 由于在一定的数据体范围内, 数据与计算之间有相关性, 在此级并行, 则需进行数据流分解, 即让协方差矩阵在线内道数据上并行计算, 最后再叠加计算结果. 通过以上两级的并行, 既可最大限度发挥多核并行计算的优势, 也可最大限度降低 I/O 延迟.

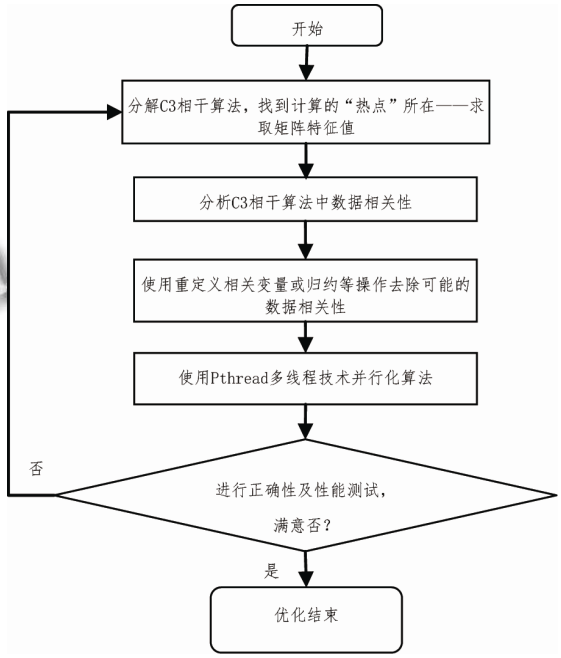

图 1 算法并行化流程设计

以下是在对串行 C3 相干算法分析的基础上, 结合并行计算的程序设计思路[9,10], 得出该 C3 相干算法并行化流程设计图, 如图 1 所示. 在算法并行化流程中,

首先解矩阵计算无关性，让计算可以不用同步机制实现地震线级数据的并行；其次，在核心计算部分使用数据流解决数据相关性，使用锁同步流操作顺序；最后把这两部分使用 Pthread 线程执行．通过持续的迭代测试和优化，达到最好的并行化性能．

实现的相干体属性提取并行化算法的伪代码如表 1 所示．

表 1　相干体属性提取并行化算法的伪代码

```
控制线程伪代码
  CThreadPool *ptp = CThreadPool(); //创建线程池管理实例
  for (i = startLine; i <= endLine; i += inc) //测线循环
  {
   ReadOneLine(); //读一条线数据
   _nThrs=nPtp->GetThreads; //获取当前系统负载平衡线程数
   AllocDataToCalcThreads(); //分配线程数据
  StartsAllCalcThreads(); //唤醒计算线程
  WaitCompleteAllThreads(); //等待所有计算线程计算完成
   ReadOneLineNext(); //读下一条线数据
  OutputResult(); //输出计算数据
  }
```

```
计算线程伪代码
  while (1)
  {
        //获取核心计算线程数
        _nCalcThrs = GetCalcThreads();
      WaitData(); //等待计算数据
      Init(); //相干体算法初始化
      if (_nCalcThrs)
          MainCalcFuncP();//执行并行相干体核心算法主体
      else
          MainCalcFunc(); //执行串行相干体核心算法主体
       if (IsMainCalcFinished()) //核心算法计算完成否
          NotifyCalComplete(); //通知控制线程计算完成
  }
```

表 1 通过控制和计算线程，实现了算法的并行化，控制线程又利用计算线程计算的时间去读取下一条线的数据，实现了读数据和计算的重叠，最后在线程池动态负载平衡的控制下，高效地发挥了多核系统的性能．

3　实验结果与分析

3.1　测试环境及数据

本文的测试环境配置为：型号 Quad-Core AMD Opteron(tm) Processor 8380，主频为 2511.615 MHZ，含 32 核的 CPU；128GB 的内存；Red Hat Enterprise Linux AS release 4(64 位操作系统)；千兆以太网卡．

测试数据为川东北某区块的三维叠偏数据，数据体共 30GB 大小；共有 2909 条线，每条线有 1760 道，每道的样点数为 1501 个．下文提到的 Amp 数据体即地震记录的振幅数据体．

3.2　测试结果及分析

图 2 给出了 Amp 数据体的相干分析分别在串行执行和并行执行下的运算总时间．

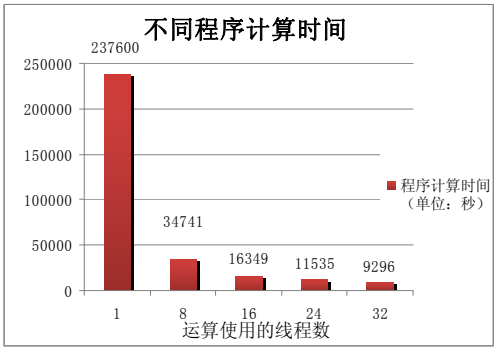

图 2　Amp 数据体在不同线程数下串行程序和并行程序的计算时间比较

图 3 给出了 Amp 数据体的并行程序在 32 核机器下，不同线程数相较于串行程序的加速比．

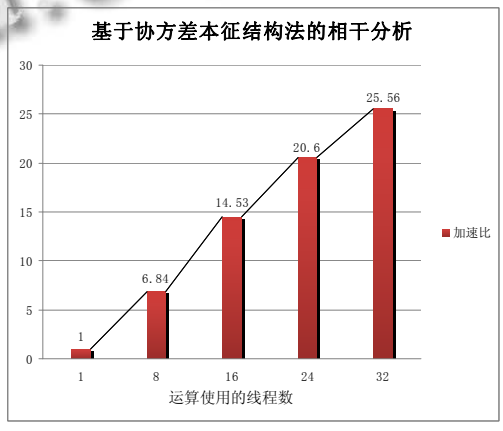

图 3　Amp 数据体并行程序在不同线程数下相较于串行程序的加速比

图 4 为 Amp 数据体并行程序在不同线程数下相较

于串行程序的并行效率.

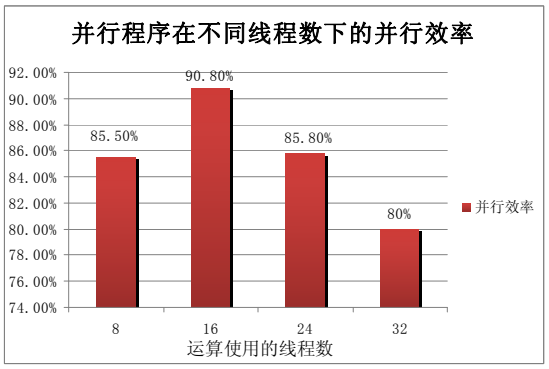

图 4 Amp 数据体并行程序在不同线程数下相较于串行程序的并行效率

由以上图可以看出: ① 随着线程数的增加, 程序的加速比是增大的; ② 随着线程数的增加, 并行效率并不都随着增加, 线程数越多, 并行效率反而降低, 在线程个数与并行加速比两者协调下能取到性价比的最佳值. 因此由图 3 加速比增长的斜率和图 4 的并行效率, 可得出经验值: 一般多线程计算时只开启计算机硬件总核数的一半线程时性价比最高.

4 结语

本文描述了基于 Pthread 的多线程并行相干体算法实现, 通过数据测试与分析, 证明基于 Pthread 的并行相干体算法能极大地提高相干体计算速度, 借助 Pthread 多线程技术和动态负载平衡可以在多核平台上有效地发挥硬件优势, 使得系统的计算性能得到极大的提升.

在相干体算法的并行化研究与实现中, 有如下一些值得借鉴的经验: (1) 测试找出整个算法中可并行化的代码段, 且一次执行此代码段的时间必须满足在此次代码段数据 I/O 的时间内, 达到 I/O 与计算重叠. (2) 由于算法使用在粗粒度机器环境上, 所以本算法选取在线方向上进行并行化, 而不是在道方向, 这样就减少了通信和等待数据 I/O 的时间. (3) 本算法并行化中, 通过控制线程和计算线程的功能划分, 达到了只使用 2 把锁, 减少了同步的开销. (4) 通过高效的线程池实现线程管理和动态负载平衡, 既可降低线程设计的复杂性, 也可提供高性能的系统应用环境.

总之, 本文中使用和讨论到的一些技术和方法, 具有并行化算法的一般性和特定性, 在类似并行化算法中, 具有很好的参考价值.

参考文献

1 OpenMP, the OpenMP ARB. http://www.openmp.org

2 POSIX Threads Programming. https://computing.llnl.gov/tutorials/pthreads/

3 Stevens WR, Rago SA. Advanced Programming in the UNIX Environment. Beijing: People's Posts and Telecommunications Press.

4 柴振友,等.相干技术在三维地震勘探构造解释中的研究与应用.物体化探计算技术,2000,22(1):30–33.

5 吴连贵,易瑜,李肯立.基于 CUDA 的地震数据相干体并行算法.计算机应用,2009,29(3):912–914.

6 Marfurt KJ, et al.周巍译.用基于相似的相干算法计算三维地震属性.石油物探译丛,1996,6(3):22–37.

7 向富强.地震相干分析和时频分析方法及其在储层描述中的应用[硕士学位论文].成都:成都理工大学,2008.8–14.

8 王大伟,刘震,等.地震相干技术的进展及其在油气勘探中的应用.地质科技情报,2005,24(2):71–76.

9 陈国良.并行计算——结构算法编程.北京:高等教育出版社,2003.

10 Akhter S, Roberts J.李宝峰等译.多核程序设计技术.北京:电子工业出版社,2007.72–240.

(上接第 37 页)

京:北京航空航天大学出版社,2002.

3 Texas Instruments. Msp430x1xx Family User's Guide. Dallas: Texas Instruments, 2004.

4 甄丽平,司绍伟.具有无线收发功能的气体检测装置设计.中国科技信息,2011(9):166－167.

5 Kaveh P. Nordic nRF905 product specificationn. Nordic Corporation, 2007.

6 Nordic VLSI ASA nRF905 Product Specification. Norway: Nordic VLSI ASA,2005.

7 李栋,张林,徐保国.无线温室信息监测系统设计.微计算机信息,2009,25(3-2):38－39,63.

8 张帅,徐伟,王克家,曹巍巍.基于 MSP430 和 LabVIEW 的温度控制系统设计.哈尔滨商业大学学报(自然科学版),2010,26(4):472－474,509.