

多核环境下潮汐分析辅助软件并行绘制技术研究

单维锋, 陈福明, 李 军

(防灾科技学院 灾害信息工程系, 河北 三河 065201)

摘要: 为方便科研人员使用 Baytap-G 潮汐分析软件, 基于 C#. NET 程序设计语言设计了一套潮汐分析辅助软件, 实现了 Baytap-G 软件的封装, 完成了输入数据格式的自动转换、输出数据的自动提取, 以及水位、振幅、相位数据的可视化展示; 在介绍 C#. NET 语言中任务 (TASK) 并行编程模型的基础上, 详细讨论了绘制振幅、相位图任务分解、并行化程序设计考虑及其实现; 实验结果表明, 合理设计并行程序可以充分利用多核计算机的计算资源, 提升程序运行效率, 但是过多的任务数、不均匀的工作负载通常会影响并行程序的效率。

关键词: 多核; 潮汐分析; 并行; 绘制; 任务

Research on Parallel Drawing Technology of Tide Analysis Aided Software Under Multi-core Environment

Shan Weifeng, Chen Fuming, Li Jun, Wang Jinfeng

(Department of Disaster Information Engineering, Institute of Disaster Prevention, Sanhe 065201, China)

Abstract: In order to help researcher to use Baytap conveniently, a new tide analysis aided software based on C#. NET programming language is designed, which encapsulates the Baytap-G software, and automatically transfer the input data into the target format. It also provides the ability of data extraction from the Baytap-G output data file and visual drawing for water level, phase and amplitude data. After introducing of TASK parallel programming model in C#. NET language, the paper analyzes the task decomposition of drawing diagram for the water level, amplitude and phase data, then gives the design and implement consideration of parallel program. The experimental results show that the reasonable design of parallel programs can make full use of multi-core computer computing resources, and improve the efficiency of the program. However, too much tasks and uneven workload usually seriously affect the efficiency of parallel program.

Keywords: multi-core; tide analysis; parallel; drawing; task

0 引言

多年来, 国内外研究人员广泛开展了潮汐应力与地震活动性关系的研究, 探索地震孕育规律^[1-4]。近年来, 开始研究固体潮的相位变化及其与孕震有关联的潮汐参数变化和地震关系研究。重力固体潮汐的研究工作主要是通过对观测数据实施调和参数分析获得相关参数 (振幅因子和相位滞后), 然后研究这些数据的变化与地震之间的关系^[5]。

Baytap-G 是国际上广泛应用的固体潮观测数据分析软件^[6], 由日本 Tamura 博士研制, Baytap-G 软件使用 Fortran 语言编写。由于 Baytap-G 软件参数众多, 科研人员经常因为参数配置问题或数据格式问题, 导致运行错误; 其次, 软件要求的输入数据格式和科研人员下载到的水位数据格式不同, 不仅需要进行数据格式转换, 而且需要将开始时间、数据记录总条数等信息同时写入到数据文件和参数配置文件中, 工作繁杂, 易于出错, 从而导致计算结果错误; 此外, 输出文件中虽然包含了各个波的振幅、相位、潮汐因子等数据, 但是除此之外还有很多其它辅助数据, 数据可读性差、波形参数数据提取

困难, 迫切需要一套辅助软件帮助科研人员形象直观的完成参数配置、输入数据格式自动转换, 数据自动填充, 并从计算结果中快速提取振幅、相位、影响因子等波形数据, 实现原始水位数据、振幅、相位数据和地震震级数据的可视化展示。

此外, 随着多核时代的到来, 传统的串行程序难以利用现有有多核计算机的 CPU 计算资源。以多线程技术为代表的并行程序设计技术能够有效地发挥多核 CPU 的计算能力, 并可显著提高程序的执行效率。虽然多线程技术在单核时代已经存在了数十年, 但只有在多核计算机或多 CPU 计算机上才真正实现了并行化。因此, 随着多核 CPU 的普及, 软件由串行编程时代开始步入并行编程时代。很多软件设计已经开始充分考虑多核资源的利用^[7]。

本文研制的 Baytap-G 潮汐分析辅助软件就是为了方便科研人员的使用, 应用 C#. NET 编程语言对 Baytap-G 软件进行了封装, 完成了输入数据格式的自动转换和输出数据的自动提取, 实现了水位、振幅、相位数据的并行可视化, 降低了使用难度。

本文在介绍潮汐分析辅助软件及 C#. NET 编程语言中基于任务的新并发编程技术的基础上, 分析了并行绘制图形关键技术, 通过对比实验分析了软件并行化后的绘制性能。

1 潮汐分析辅助软件并行化分析

Baytap-G 潮汐分析辅助软件采用 SDI 程序风格, 包含“系统”、“潮汐数据处理”和“关于”3 个主菜单项, 如图 1 所示。“系统”菜单项下面包含“退出”子菜单项。“潮汐数据

收稿日期: 2017-02-14; 修回日期: 2017-03-07。

基金项目: 中国地震局教师科研基金 (20130111); 国家自然科学基金 (41504037); 地震科技星火计划 (XH16059); 中央高校基本科研业务费专项资金 (创新项目团队资助计划) (ZY20160106)。

作者简介: 单维锋 (1977-), 男, 山东在平人, 博士, 副教授, 主要从事地震大数据、可视化技术方向的研究。

处理”下包含“潮汐分析”、“绘制水位图”、“绘制振幅相位图”等功能, 功能较为简单。



图 1 Baytap-G 潮汐分析辅助软件

“潮汐分析”功能是对 Baytap-G 软件的封装, 首先, 软件将用户在本界面录入的各种参数信息以及选择的水位数据文件, 按照 Baytap-G 软件格式保存到参数文件 input05.dat、input12.dat 和 input14.dat (如果包含气压数据) 文件中。然后, 调用 baytap-G 软件进行计算, 计算完成后, 会自动从 baytap-G 软件输出文件 output06.dat 中提取各个波形数据, 保存到用户指定的输出文件夹下, 每个波形数据保存为一个文件, 命名为 Sheet_xxx.txt, 其中 xxx 表示波的名称如 O1、M2、N2 等, 最多 12 个波。在该文件中, 每行为一条记录, 包含了时间、振幅、相位、误差等数据, 中间用逗号分隔, 时间仅记录每次潮汐计算的开始时间。

通过对 Baytap-G 程序进行分析, 发现如果输入的水位数据多, SPAN 和 SHIFT 参数设置较小情况下, 计算工作量会非常大。但由于 Baytap-G 程序每个 SPAN 数据的计算依赖于用户选择的起始数据和以前的计算结果, 因此难以并行化。

此外, 由于“绘制水位图”是对用户待处理水位数据的简单可视化实现, 计算任务量并不大, 因此也没有必要进行并行化。程序运行界面如图 2 所示。

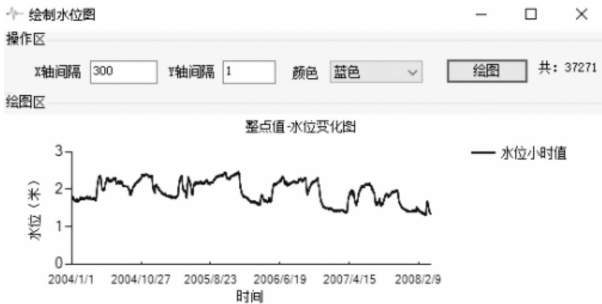


图 2 水位图

绘制振幅、相位图是为了直观形象的对比分析各个波形数据和水位数据及地震之间的关系, 如图 3 所示。用户点击“选择数据文件夹”后面的浏览按钮, 选择存放 baytap-G 输出数据的文件夹, 然后直接点击“绘制”按钮, 就可以同时将水位数据、振幅数据和相位数据绘制在界面上。用户可以通过选择 SYMBOL 后面下拉框的值绘制不同的波形数据图, 用户也可以定制曲线的颜色, 以及绘制的刻度数等参数, 调整图形。水位数据、振幅数据和相位数据使用相同的时间轴, 以便进行对比, 但是 X 轴刻度可以分别进行调整。在水位数据中, 仅将

缺数数据简单置为 0, 因此从图 3 可以看出水位有多处突升和突降。图 4 的下半部分是振幅、相位数据, 由于其意义和取值范围存在很大差别, 因此将右侧作为振幅的 Y 轴, 左侧作为相位的 Y 轴, 共用一个 X 时间轴, 根据实际取值等比缩放放在一个相同的 Y 轴空间内。通过此图, 可以直观的观察水位数据和波形数据之间的关系。

此外, 如果在图 3 中点击“选择震级 Excel”标签后的“浏览”按钮, 指定了地震目录数据, 还可以根据地震的发震时刻, 将震级数据同时绘制在此界面上, 每个震级用下箭头表示。地震目录 Excel 文件要依次包含发震日期、年、月、日、时间、震级、经度、纬度、深度等信息, 否则无法正常标注。

由此可见, 振幅相位图绘制功能计算任务繁重, 如果采用传统的串行编程模式, 一方面难以发挥多核计算机的计算能力, 另一方面由于串行计算时间长, 用户需要更长的等待时间。在确定了绘制坐标原点、绘制长度和高度之后, 完全可以并行绘制水位图、振幅图和相位图, 因为这些任务之间几乎没有依赖关系。

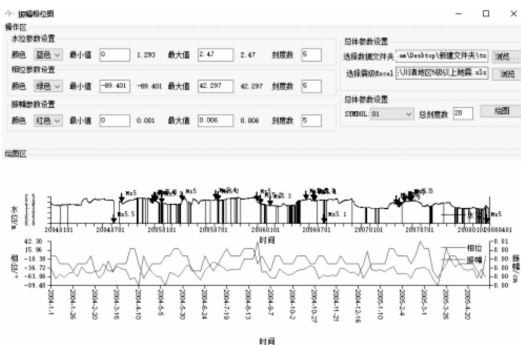


图 3 振幅相位图

2 C#.NET 并程序序设计技术

随着多核时代的到来, 传统的串行程序难以利用现有计算机的多核 CPU 计算资源, 多线程技术可以有效地发挥多核 CPU 的计算能力, 在很大程度上提高程序的运行效率。虽然多线程技术在单核时代已经存在了数十年, 但是多线程在实际执行时, 在同一个时刻只有一个线程处于活动状态。而在多核时代, 多个线程可以同时并行执行。由于多核体系结构和单核体系结构在缓存和线程优先级等方面存在巨大的差别, 在设计多线程应用程序时, 需要仔细分配各个线程的工作负载才能达到最高的性能。众所周知, 在基于多线程技术的并行程序设计中, 软件开发者必须通过锁、信号量等同步机制来进行细粒度的控制, 导致开发难度大、难以理解、难以检验等诸多问题。

微软在发布的 .NET Framework 4.0 中增强了其并行编程功能, 引入了一个新的概念——任务。作为并行计算的重要组成部分, 任务是对传统线程池技术的一个补充和完善。众所周知, 在使用线程池技术时, 一旦把要执行的任务放进去后, 就无法通过内置的方式知晓任务何时才能执行完成。任务相比线程池有更小的开销和精确的控制方式, 这使得多核环境下并行编程变得相对简单。

在 .NET Framework 4.0 的中 System.Threading.Tasks 命名空间下包含了任务相关类, 如果执行的任务不需要返回值, 可以使用 Task 类, 若需要处理返回值, 可以使用 Task<

TResult>泛型版本。

有两种方式创建任务，一种是通过 Task.Factory.StartNew 方法来创建一个新任务，任务创建完成，其中的代码被马上执行。

另一种方法是通过 Task 类的构造函数来创建一个新任务，如：

```
Task task = new Task ( () => Console.WriteLine ("Test Task!");
```

用此方法时，只是把要完成的工作交给任务，但任务并未真正开始执行。需要调用任务对象 task 的 Start 方法，任务才会在将来某个时候开始执行。

此外，还可以调用 Task 类的 WaitAll 或 WaitAny 方法来等待一组或任意一个任务的完成，也可以调用某个具体任务的 IsCompleted 属性来判断任务是否完成。

由此可见，相对于线程池技术，任务提供了更易于理解的封装，具有更易用的访问接口，特别适合任务并行等同步操作较少的场合。

3 基于任务的振幅相位图并行绘制技术

3.1 初步任务分解

并行绘制振幅相位图，首先是对数据进行预处理，把水位数据和震级数据合并为一个内存结构中（如链表），在处理过程中顺便求出水位的最大值和最小值，以便等比例缩放水位数据，同时将缺数数据置为 0。然后根据选择的波，读取波形数据文件中的时间、振幅、相位数据，并分别找出振幅和相位数据的最大值和最小值，以便计算并绘制高度。接着需要定义水位、震级图（图 3 上半部分）坐标原点、高度和宽度，定义振幅、相位图（图 3 下半部分）的坐标原点、高度和宽度，其中上、下两部分图的宽度相同，以便对比分析。上述任务完成后，就可以绘制坐标轴、水位、振幅、相位了。所以一种可能的任务分解如图 5 所示。

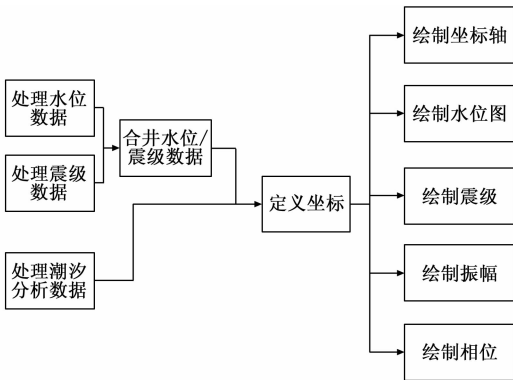


图 4 初步任务分解图

3.2 并行任务实现

虽然可以将图 4 中的每个任务直接映射到 C#.NET 中任务，从而实现并行化软件改造，但是由于任务最终通过线程技术实现，因此任务太多，导致线程调度过于频繁，无谓的消耗会影响系统的性能。此外，还需要考虑任务之间的负载均衡问题，系统的运行时间取决于最后任务执行完毕的时间。

因此，考虑到震级的数据量远远小于水位数据，将绘制水位图和绘制震级任务合并为一个任务。另外，由于潮汐计算产

生的振幅、相位数据依赖于 SHIFT 参数的设置，数据量约为水位/SHIFT，因此远小于水位数据个数，因此将绘制振幅和相位图、绘制坐标任务合并一个新的任务。此外，将定义坐标及其之前的所有任务合并为一个并行任务。设计完成后，程序实际的并行任务模型如图 5 所示，仅包含 3 个任务。

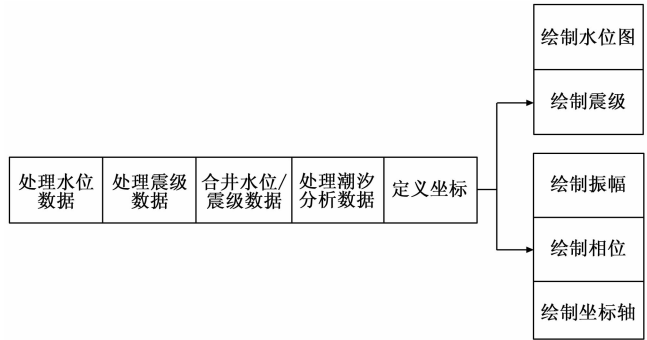


图 5 任务执行图

将上述每个功能任务封装为一个方法，然后通过 Task.Factory.StartNew 方法创建并执行一个任务，核心代码如下：

```
Task t1 = Task.Factory.StartNew(() =>
{
    处理水位数据;
    处理震级数据;
    合并水位/震级数据;
    处理潮汐分析数据;
    定义坐标;
});
Task t2 = t1.ContinueWith(t =>
{
    绘制水位图;
    绘制震级;
});
Task t3 = t1.ContinueWith(t =>
{
    绘制振幅图;
    绘制相位图;
});
```

在上述代码中任务象 t1 的 ContinueWith 方法表示等待 t1 任务完成后再执行方法主体部分的代码，即任务 t1 执行完成后才会执行 t2 和 t3 任务，所以 t1 和 t2 之间是串行执行，t1 和 t3 之间也是串行执行，但是 t2 和 t3 是并行执行的。

4 实验结果分析

为了对比潮汐分析辅助软件并行化前后系统的性能，通过本软件的潮汐分析功能产生了 7 组潮汐分析数据。绘图数据包括水位、振幅、相位数据。理论上，并行绘制技术涉及到线程的创建、调度等额外开销，在处理的数据规模较大和任务分配较为平均的情况下，会有较大的性能提升。因此，考虑到各种情况，选取测试用例如表 1 所示。

T5 测试用例数据最少，T4 用例的水位数据最多，T6 和 T7 测试了相同水位个数下不同振幅/相位数的情况。各测试用例的结果如图 7 所示，每个用例分别运行 5 次，然后取平均运行时间作为该用例的耗时时间。