

# 基于 OpenMP 的 Kriging 插值算法研究

陈欢 谢健

(湖南大学软件学院 长沙 410082)

**摘要** 随着多核处理器的普及,并为了充分利用多核 PC 机的特性,计算机技术逐渐向多核架构及多核计算技术发展。为提高对湖南地区 100m×100m 小网格气温插值的速度,采用以 OpenMP 为标准的基于共享存储的并行编程模型对 Kriging 插值算法进行改进。在不同核的多核 PC 机中,采用 100m×100m 小网格和 500m×500m 小网格地形数据对平均气温进行插值,不仅有效减少了插值时间和提高了算法的加速比,而且集成到业务系统中大大提升了系统的反应时间及性能。

**关键词** 多核计算, OpenMP 模型, Kriging 插值, 小网格插值

**中图分类号** TP391 **文献标识码** A

## Kriging Interpolation Algorithm Based on OpenMP

CHEN Huan XIE Jian

(College of Information Science and Engineering, Hunan University, Changsha 410082, China)

**Abstract** With the popularity of multi-core processors, full use of multi-core PC's performance, computer technology is increasingly to multi-core architecture and multi-core computing technology development. To improve the small Hunan 100m × 100m grid interpolation rate of temperature, using the OpenMP standard for shared memory based parallel programming model to achieve the kriging interpolation algorithm based on OpenMP. Different multi-core PC, using a small grid 100m × 100m and 500m × 500m grid terrain data for a small average temperature for interpolation, not only effectively reduce the time and interpolation algorithms to improve the speedup, and significantly integrated into business systems enhance the system's response time and performance.

**Keywords** Multi-core computing, OpenMP model, Kriging interpolation, Small grid interpolation

## 1 引言

由于气候变化的加剧,全球气候变暖是不争的事实。随着全球气候变化科学的发展,迫切需要气候要素的空间动态数据集,并且所要求的时间和空间分辨率也越来越高。气温是基本气候因子之一,平均气温是进行热量资源分析、自然区划和指导农业生产及计算农业生产潜力的重要参数。某一区域的气温数据一般是通过该区域若干具有代表性的气象观测站观测获得,而气象站数量有限。利用区域内有限的观测站点的气温数据,对其它空间格点的数据进行插值,才能获得整个区域或某部分区域的近似气温分布,以进行气候分析与研究。

空间插值方法有两种分类方法:一种为整体插值法(global methods)和局部插值法(local methods)两类;另一种是将插值方法归结为整体插值法、局部插值法、地学统计法(statistical methods)和混合插值法(mixed methods)四类<sup>[1]</sup>。克里金(Kriging)插值法是局部插值法,是一种最优内插法<sup>[2]</sup>。Kriging 插值是建立在变异函数空间分析基础上,对有限区域内的区域化变量取值进行无偏最优估计的一种方

法<sup>[2]</sup>。与其它插值方法相比,其不仅考虑了待插点与邻近已知点的空间相关性,而且能够给出估计误差。

用该方法对湖南地区 100m×100m 小网格点(有效格点约为 34520240)气温数据进行海量插值,由于小网格点个数达到  $10^7$ ,级导致计算量较大、计算时间较长。其虽然在高性能计算机<sup>[3]</sup>或者分布式计算机<sup>[4]</sup>中得以解决,但是在 PC 中无法满足实时计算的要求。

随着多核处理器的普及,多核计算技术得到发展,其在数据库、创作 3D 图像、数学分析和网络服务等多线程多任务的计算需求领域发挥着极大的作用<sup>[5]</sup>。为解决以上问题,提出了基于 OpenM 的共享存储的多核并行 Kriging 插值算法并得到实际应用,满足了对湖南地区小网格实时插值的需求。

## 2 OpenMP

串行编程模型和并行架构的矛盾日益突出,需要根据多核的平台架构来设计一种并行编程模型,使基于新的编程模型的程序可以在多核平台上执行效率更高,性能更优越<sup>[6,8]</sup>。

OpenMP 起源于 ANSI X3H5 标准,是共享存储系统编程的一个工业标准。OpenMP 是在共享存储的多核体系结构

陈欢(1985—),男,硕士生,主要研究方向为多核计算、高性能计算, E-mail: huan2lily@gmail.com; 谢健(1963—),男,硕士,副研究员,主要研究方向为计算机应用软件、高性能计算。

上对多核并行编程模型应用的编程接口 API<sup>[6-9]</sup>。OpenMP 程序编程模型包括编译指导、运行函数库和环境变量等,可以显式地指导编译器如何以及何时执行应用程序中的并行部分。

如图 1 所示,OpenMP 采用 fork-join 模型。一个 OpenMP 程序从单个线程开始执行,在程序的某些点遇到指导指令需要并行执行时,程序派生出额外的线程,组成一个线程组。这些线程在并行区域的代码区中并行执行。线程到达并行区域的末尾时等待,直到整个线程组都到达,然后它们连接在一起,只有初始或者主线程继续执行,直到下一个并行区域(或者程序结束)。与 MPI 相比,OpenMP 在串行程序中可并行部分插入指导指令而并行执行,具有简单、移植性好和可扩展性等优点<sup>[10]</sup>。

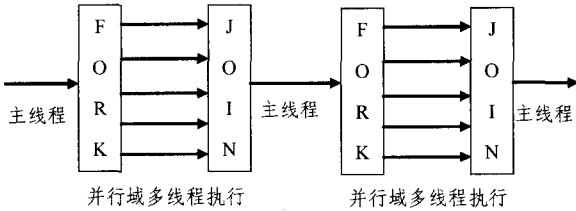


图 1 fork-join 模型

### 3 基于 OpenMP 的并行 Kriging 算法

#### 3.1 Kriging 算法

Kriging 法是一种地质统计方法,是一种最优、线性、无偏内插估计量,由法国的统计学家 Matheron 提出,在 1951 年首次将统计学技术运用到地矿评估中<sup>[11]</sup>。Kriging 插值的核心技术就是用半变异函数模型代表空间中随距离变化的函数,在无偏估计与最小估计变异数的条件下,决定各采样点的权重系数,最后以各采样点与已求得的权重线性组合来求解空间任意点的内插估计值<sup>[13]</sup>。

记  $Z(X)$  为区域化变量,设  $Z(X_i)$  为  $m$  个采样点  $X_i (i=1, 2, 3, \dots, m)$  的属性值,  $X_i = (x_i, y_i)$  是采样点在网格点对应的坐标。根据 Kriging 插值原理,待估点  $X_0$  处的属性值  $Z(X_0)$  是  $m$  个已知采样点的属性值的加权和:

$$Z(X_0) = \sum_{i=1}^m \lambda_i Z(X_i) \quad (1)$$

式中,  $\lambda_i (i=1, 2, \dots, m)$  为待求权重系数。一般可以采用:

$$\gamma(X_i - Y_j) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} (Z(X_i) - Z(X_j))^2 \quad (2)$$

通过实验获得半变异数,并与选定的理论模型(球状模型、指数模型等)进行参数拟合。其中,  $(X_i - Y_j)$  表示两个观测站  $X_i$  与  $X_j$  之间的距离  $h$ ,  $\gamma(X_i - Y_j)$  表示与  $h$  对应的变异函数,  $Z(X_i)$  与  $Z(X_j)$  分别为与观测站对应的属性值,  $N(h)$  表示观测站之间距离为  $h$  时观测站的计数。

限定待估点  $Z^*(X_0)$  与真正的  $Z(X_0)$  具有相同的数学期望,满足无偏性要求;同时按照无偏条件下使估计方差达到最小的准则,则有 Kriging 系统方程式如下:

$$\begin{cases} \sum_{i=1}^m \lambda_{0i} \gamma(X_i - Y_j) + \mu = \gamma(X_0 - X_i) \\ \sum_{i=1}^m \lambda_{0i} = 1, i=1, 2, \dots, m \end{cases} \quad (3)$$

式中,  $\mu$  为拉格朗日算子。在求出权系数  $\lambda_i (i=1, 2, \dots, m)$  后,就可求出待估点  $X_0$  处的属性值  $Z^*(X_0)$ 。

$$\text{若令 } \gamma_j = \gamma(X_i - Y_j), \text{ 式(3)用矩阵表示为 } A \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \gamma_0,$$

则可以得到权系数的求解方程如下所示:

$$\begin{bmatrix} \lambda \\ \mu \end{bmatrix} = A^{-1} \gamma_0 \quad (4)$$

其中,

$$\begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \dots \\ \lambda_m \\ \mu \end{bmatrix}, \gamma_0 = \begin{bmatrix} \gamma_{01} \\ \dots \\ \gamma_{0m} \\ 1 \end{bmatrix}, A = \begin{bmatrix} \gamma_{11} & \dots & \gamma_{1m} & 1 \\ \dots & \dots & \dots & 1 \\ \gamma_{m1} & \gamma_{m2} & \gamma_{mm} & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad (5)$$

矩阵  $A$  中的  $\gamma_{ij}$  为各地面观测站值之间的半方差值,矩阵  $\gamma_0$  中的  $\gamma_{0i}$  为第  $i$  个观测站与待插值格点的半方差值。

在插值处理过程中,网格地形数据中存在一定数量的缺省值,这部分数据是不需要进行气温插值的。Kriging 串行算法如下:

- 1) 读小网格地形数据、日平均气温数据及 97 个观测站经纬坐标数据;
- 2) 根据观察站及日平均气温数据,计算变异函数和拟合变异函数  $\gamma(X_i - Y_j)$ ;
- 3) Kriging 插值过程:
 

```
for i=1 to N
    for j=1 to M
        for k=1 to P
            计算小网格点到观测站的距离;
        end for
        for k=1 to P+1
            for l=1 to P+1
                计算观测站之间的协方差并确定系数矩阵;
            end for
        end for
        for k=1 to P
            计算待插小网格的值;
        end for
    end for
end for
```
- 4) 存储插值结果。

其中,  $N \times M$  是湖南地区  $100\text{m} \times 100\text{m}$  的有效小网格数,  $P$  为观测站数,湖南地区有 97 个观测站。

串行 Kriging 算法比较复杂,且循环较多导致计算量大,在性能比较好的单核 PC 机上计算时间达到几分钟甚至十多分钟,等待时间较长。

#### 3.2 基于 OpenMP 的并行 Kriging 算法

OpenMP 并行化一般针对循环级的细粒度的并行化,是基于共享任务的。在依赖性较小、数据相对独立的循环迭代部分前插入编译指导指令,将它所包含的代码划分给线程组的各成员来执行。在串行 Kriging 算法中,绝大多数的主要计算量都集中在循环中。由于线程和进程反复切换及线程不

断地产生和终止、线程的私有变量大量的内存消耗带来的性能影响,因此它在优化负载平衡和优化内存利用之间进行折中,选择计算时间占全局计算时间比例大的循环进行并行化。

串行 Kriging 算法主要计算时间耗在插值过程的四层循环内。由于在特定的小网格插值计算环境下,每个点的计算都是可以独立完成的,相互之间没有数据相关性,因此可对其进行优化。串行 Kriging 算法中第 3 步插值过程可修改如下:

```
(void) omp_set_nested(TRUE);
#pragma omp parallel for default(none)
private(i,j) \
schedule(dynamic)
for i=1 to N
    for j=1 to M
        # pragma omp for private(k,distance)
        for k=1 to P
            计算小网格点到观测站的距离(distance);
        end for
        # pragma omp for private(k,matrix)
        for k=1 to P+1
            for l=1 to P+1
                计算观测站之间的协方差并确定系数矩阵(matrix);
            end for
        end for
        # pragma omp for private(k,value)
        for k=1 to P
            计算待插小网格的值(value);
        end for
    end for
end for
```

#### 4 性能分析

在实验中,分别采用湖南地区 100m×100m 小网格和 500m×500m 小网格地形数据对某日的平均气温数据进行插值,矩阵大小分别为 5911×5840 和 1256×1145。变异函数模型采用球状模型,用附近 8 个观测站的数据对小网格进行插值,分别使用单核 PC 机、双核 PC 机、四核 PC 机及八核 PC 机对算法进行测试并使用 Intel 的 Parallel Studio 工具不断调优。测试结果如表 1 和表 2 所列。

表 1 100m×100m 小网格插值效率比较

| 处理器核数 | 计算时间/s | 加速比  | 效率    |
|-------|--------|------|-------|
| 单核    | 509.31 | —    | —     |
| 双核    | 287.57 | 1.77 | 0.885 |
| 四核    | 156.21 | 3.26 | 0.815 |
| 八核    | 94.01  | 5.42 | 0.678 |

表 2 500m×500m 小网格插值效率比较

| 处理器核数 | 计算时间/s | 加速比  | 效率   |
|-------|--------|------|------|
| 单核    | 199.31 | —    | —    |
| 双核    | 127.17 | 1.57 | 0.78 |
| 四核    | 70.71  | 2.82 | 0.70 |
| 八核    | 43.33  | 4.60 | 0.57 |

从表 1 和表 2 可以看出,基于 OpenMP 的并行 Kriging

算法随着核数的增加,计算速度也大大提高,计算时间显著减少,并且在单核到双核和双核到四核的计算时间都减少一倍,在多核 PC 机环境中都达到了较好的并行效果。

然而加速比曲线图如图 2 和图 3 并不是直线递增图,因为随着核数的递增,较多的线程的产生及线程之间的数据交换增加了时间开销,并行效率有所下降。在计算时间、加速比和效率分析上都验证了基于 OpenMP 的 Kriging 算法都能达到了较理想的效果。

为充分利用多核处理器的特性,将基于 OpenMP 的并行 Kriging 插值算法应用到气象相关业务系统中,不仅提高了计算速度和业务系统的反应时间,而且缩短了用户等待时间,提高了业务系统的整体性能。

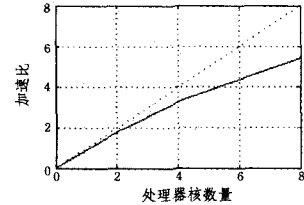


图 2 100m×100m 小网格插值加速比曲线图

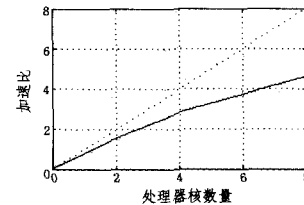


图 3 500m×500m 小网格插值加速比曲线图

**结束语** 针对湖南地区 100m×100m 小网格进行插值,提出了基于 OpenMP 标准的 Kriging 插值算法,在多核 PC 机中有效地提高了该算法的效率,实现了多核并行技术的应用。

对气温进行小网格插值计算时,由于各个网格点插值计算时相互之间没有数据相关性,具有良好的独立性,从而能有效地应用基于 OpenMP 标准的 Kriging 算法。

在信息化业务系统的建设中采用多核并行计算技术,可以极大地减少系统的数据处理时间,提高系统反应速度,进而提高系统的整体性能,使得在 PC 机中处理大量数据成为可能。这对业务系统如何有效地使用多核并行计算技术提供了有价值的参考。

#### 参考文献

- [1] 何红艳,郭志华,肖文发.降水空间插值技术的研究进展[J].生态学杂志,2005,24(10):1187-1191
- [2] Kerry K E, Hawick K A. Kriging interpolation on high-performance computers[J]. High-Performance Computing and Networking, 1998, 1401:429-438
- [3] Kerry K E, Hawick K A. Spatial Interpolation on Distributed High-Performance Computers[R]. DHPC-015. Australia: University of Adelaide, 1997
- [4] Kenjiro T, Kenji K, Toshio E, et al. A Parallel Programming Model for Accommodating Dynamically Joining/Leaving Re-

sources[C]//Proc. of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, USA;[s. n.], 2003

[5] 李晓明,王韬,刘东,等. 走进多核时代[J]. 计算机科学与探索, 2008

[6] Chandra R, Dagum L, Kohr D, et al. Parallel Programming in OpenMP[M]. Morgan Kaufmann Publishers, 2001

[7] Klemm M, Bezold M, Gabriel S, et al. Reparallelization techniques for migrating OpenMP codes in computational grids[J]. Special Issue: The Best of CCGrid'2007: A Snapshot of an 'Adolescent' Area, 2009. Concurrency and Computation: Practice and Experience, 2009, 21: 281-299

[8] Shah S, Haab G, Petersen P, et al. Flexible control structures for parallelism in OpenMP[J]. Concurrency: Practice and Expe-

rience, 2000; 1219-1239

[9] Schmidl D, Terboven C, Mey D, et al. Binding Nested OpenMP Programs on Hierarchical Memory Architectures[J]. Beyond Loop Level Parallelism in Openmp; Accelerators, Tasking and More, 2010, 6132

[10] Rabenseifner R, Hager G, Jost G. Hybrid MPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes[J]. Parallel, Distributed and Network-based Processing, 2009. High Performance Comput. Center Stuttgart (HLRS), 2009(2): 427-436

[11] 王家华,周叶,高海余. 克里金地质绘图技术[M]. 北京:石油工业出版社, 1999

[12] 曾怀生,黄生享. 基于 Kriging 方法的空间数据插值研究[J]. 测绘工程, 2007, 16(5)

(上接第 382 页)

如图 2、图 3 所示,随着博弈过程的进行,各虚拟机资源份额和效益函数值曲线趋向平稳。当处于纳什均衡点时,各虚拟机效益函数值达到最大。在实验二,采取两台相同效益函数虚拟机进行博弈,效益函数调节因子设为 10。由于效益函数一样,根据第 3 节可知,各虚拟机应该获取相同的资源,图 4 验证了此结论,当处于纳什均衡点时,系统等分资源。

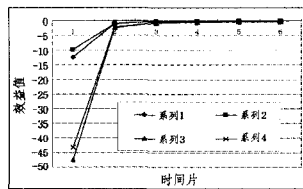


图 3 各虚拟机效益值

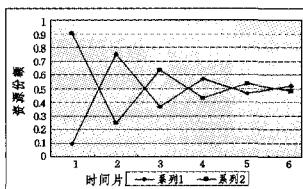


图 4 两个等效效益函数博弈

图 5 表示,虽然两个虚拟机的初始效益值不同,但是他们的效益函数一样,当处于纳什均衡时,各虚拟机的效益值都几乎相同。

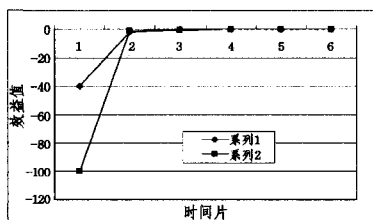


图 5 效益值

**结束语** 本文对虚拟机环境下的资源管理进行了研究,提出了非合作博弈标价模型。该模型核心部分是各虚拟机效益函数的设计。本文在参与者效益函数零点无定义下,给出了模型纳什均衡点存在性和唯一性的证明。同时通过仿真实验,验证了系统处于纳什均衡点时,各虚拟机博弈参与者的效益最大,而且资源分配方案也是公平的。通过本文标价模型分配虚拟机资源,能有效提高系统资源的利用率,节约系统成本。

## 参考文献

[1] Padala P, Hou K Y, Shin K G, et al. Automated control of multiple virtualized resources[C]//Proceedings of the 4th ACM Eu-

ropean Conference on Computer Systems (EuroSys 2009). New York: ACM, 2009; 13-26

[2] 金海,邓莉,吴松,等. 一种能耗感知的虚拟集群 CPU 资源自动再配置方法[J]. 计算机研究与发展, 2011, 48(7): 1123-1133

[3] Zhao Wei-ming, Wang Zhen-lin, Luo Ying-wei. Dynamic memory balancing for virtual machines[C]// Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual Execution Environments (VEE 2009). New York: ACM, 2009, 43(3): 37-47

[4] 李运发,徐向华,万健. 基于虚拟机负载迁移的资源调度机制[J]. 华中科技大学学报:自然科学版, 2009, 37(9): 45-48

[5] Ghosh P, Roy N, Das S K, et al. A game theory based pricing strategy for job allocation in mobile grids[C]// Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS 2004). Santa Fe, New Mexico: IEEE, 2007; 80-92

[6] Altman E, Borkar V S, Kherani A A. Optimal random access in networks with two-way traffic[C]// 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2004), 2004; 609-613

[7] Lin T H, Chatterjee M, Basu K, et al. ARC: An integrated admission and rate control framework for competitive wireless CDMA data networks using non-cooperative games[J]. IEEE Transactions on Mobile Computation, 2005, 4(3): 243-258

[8] Facchinei F, Kanzow C. Generalized nash equilibrium problems[J]. A Quarterly Journal of Operations, 2007, 5(3): 173-210

[9] Ardagna D, Panicucci B, Passacantando M. A game theoretic formulation of the service provisioning problem in cloud systems [C]// Proceedings of the 20th international conference on World Wide Web (WWW 2011). New York: ACM, 2011; 177-186

[10] Yolken B, Bambos N. Game based capacity allocation for utility computing environments[J]. Telecommunication Systems, 2010, 47(1/2): 165-181

[11] Altman E, Boulogne T, El-Azouzi R, et al. A survey on networking games in telecommunications [J]. Computers & Operations Research, 2006, 33(2): 286-311

[12] 李明欣,陈山枝,谢东亮,等. 异构无线网络中基于非合作博弈论的资源分配和接入控制[J]. 软件学报, 2008, 21(8): 2037-2049