模糊 C 均值聚类算法的并行化研究 *

张建强,郑晓薇,吴华平
(辽宁师范大学 计算机与信息技术学院,辽宁 大连 116081)

摘 要: 使用 Intel Parallel Amplifier 高性能工具, 针对模糊 C 均值聚类算法在多核平台的性能问题,找出串行程序的热点和并发性,提出并行化设计方案。基于 Intel 并行库 TBB(线程构建模块)和 OpenMP 运行时库函数,对多核平台下的串行程序进行循环并行化和任务分配的并行化设计。

关键词: 多核;并行化;模糊 C 均值算法;Intel Parallel Amplifier;OpenMP

中图分类号: TP311 **文献标识码**: A **文章编号**: 1674-7720(2010)23-0008-03

Research on fuzzy C-means clustering algorithm parallel

ZHANG Jian Qiang, ZHENG Xiao Wei, WU Hua Ping
(College of Computer and Information Technology, Liaoning Normal University, Dalian 116081, China)

Abstract: Study of fuzzy C-means clustering algorithm in multi-core platform performance bottleneck with the help of the Intel Parallel Amplifier high-performance tool, to find hotspot and concurrency, the solution of parallel design was proposed. Based on Intel TBB(Thread Building Blocks) and OpenMP, design the cycle and task distribution of the serial program in multi-core platforms.

Key words: multi-core; parallel; fuzzy C-means algorith; Intel Parallel Amplifier; OpenMP

多核处理器的迅速发展,使得多核化不断全面普及。为了应对计算机硬件的发展要求,尽可能利用多核资源,就要设计出相应的并行化应用程序。多核平台下的并行化有多种方案,利用英特尔推出的高性能分析工具 Intel Parallel Amplifier 对串行应用程序进行性能分析,寻出热点实现并行化是其中的一种方法。

模糊 C 均值聚类算法(FCM)是一种常用的聚类算法,在大规模数据分析、数据挖掘、模式识别、图像处理等领域有着非常广泛的应用。它是给定分类数,通过优化目标函数得到样本点对聚类中心的隶属度,寻找样本点的最佳分类方案。本文将多核技术应用到模糊 C 均值聚类并行算法的设计中,把目标函数迭代的过程和处理数据的过程并行化,提高聚类过程的效率及多核处理器的利用率。实验结果表明,本方法减少了程序的运行时间,显示了多核编程的高效性。

1 模糊 C 均值聚类算法(FCM)

模糊 C 均值聚类算法[1]的基本思想是确定每个样本数据隶属于某个聚类的程度,把隶属程度相似的样本数据归为一个聚类。FCM 把 n 个样本集合 $X=\{x_1, x_2, \ldots, x_n\}$ 分为 c 个模糊组,并且求每组的聚类中心 $C_i(i=1,2,\cdots c)$,使得目标函数最小,该算法是优化目标函数的迭代过程。这个过程从一个随机的隶属度矩阵开始,确定聚类中心计算目标函数,通过迭代过程达到样本分类。

初始化:给定样本数 n,聚类数 $c\in[2,n]$,模糊度 $m=2$,迭代停止阈值 ω。

(1)初始化隶属度矩阵 U,满足 $\sum_{i=1}^{c} U_{ij}=1$, $\forall j=1-n$ 对样本数据 $\mathrm{data}(i,j)$ 做标准化处理,满足:

$$\mathrm{data}'(i,j)=\frac{\mathrm{data}(i,j)-\min\ \mathrm{data}(j)}{\max\ \mathrm{data}(j)-\min\ \mathrm{data}(j)} \tag{1}$$

(2)计算聚类中心 $C_i,(i=1,\cdots,c)$, $C_i=\dfrac{u_{ij}\ \mathrm{data}'_{ij}}{\sum_{j=1}^{n} u_{ij}}$ (2)

(3) 对模糊 C 均值进行迭代运算,返回目标函数的值。目标函数的形式为:

$$T(U,c_1\cdots c_c)=\sum_{i=1}^{c} Ti=\sum_{i=1}^{c}\sum_{j}^{n} u_{ij}^{m}\ d_{ij}^{2}, \tag{3}$$

其中 $d_{ij}=\|c_i-\mathrm{data}(i,j)\|$ 为聚类中心与样本点的欧几里得距离。

* 基金项目:国家自然科学基金项目(项目编号:60603047)

(4)如果目标函数的改变量小于 ω,停止算法,否者重复(2)直到改变量小于 ω。为了确保 FMC 得到一个最优解,要不断调整隶属度矩阵,需多次运行该算法。

2 多核技术与工具软件

2.1 Intel Parallel Amplifier 高性能工具

Intel Parallel Amplifier 是英特尔在 2009 年发布的高性能工具[2],界面设计友好,操作简单方便。开发人员只需要运行工具就可对串行程序进行分析,研究分析结果进行并行化设计,确保多核的完全利用。IPA(Intel Parallel Amplifier)有以下三种类型的性能分析。

(1)热点(Hotspot)分析:运行热点分析可收集到不同类型的数据,确定应用程序运行消耗的时间,以及识别出最耗时的函数。在执行程序时,IPA 通过数据收集器定期采样,并在操作系统的协作下中断程序收集数据。它通过获取整个程序各个 CPU 核心的指令指针(IP)采样,计算出每个函数的运行时间,再用调用栈采样为程序创建调用关系树。

(2)并发性(Concurrency)分析:运行并发性分析可确定应用程序是否有效地利用了所有可执行核,识别出最有可能并行化的串行代码。它与热点分析一样收集数据信息,但是要比热点分析多,除了一般的程序运行数据,还有所有可执行核的工作情况。最理想的情况是执行程序的线程数等于处理器的可执行核数,也就是完全利用(Fully Utilized)。

(3)锁定和等待(Locks and Waits)分析:在前两种分析的基础上,运行锁定和等待分析,可获得更多的程序运行数据。

为了测试程序并行优化的效果,IPA 提供了"比较结果(Compare Results)"的功能,用来比较串行程序和并行程序性能差别。

2.2 TBB 线程构建模块

TBB 线程构建模块(Intel Thread Building Blocks)是基于 GPLv2 开源的、用来实现并行语义的 C++模板库[3]。TBB 提供了高性能可扩展的算法,面向任务编程,支持任何 ISO C++编译器,具有很好的可移植性。本文将 Intel 并行库 TBB 的 tbb_block_rang2d 和 tbb_parallel_for 配合使用,前者的作用是对一个二维的半开区间进行可递归的粒度划分;后者的作用可以实现负载均衡的并行执行固定数目独立循环迭代体。

2.3 OpenMP 并行编程模型

OpenMP 是为共享内存以及分布式共享内存设计的多线程并行编程应用接口,包含了一套编译语句以及一个函数库,是一个编译指令和库函数的集合[4]。OpenMP 也可以用于多核处理器并行程序设计中。在 OpenMP 中线程的创建是通过编译指导语句实现的,本文采用 sections 和 section 命令。sections 被称作工作分区编码,它定义了一个工作分区,然后由 section 将工作区划分成几个不同的工作段,每个工作段都由多核处理器的每个执行核并行执行。

3 C 均值聚类算法的并行优化设计

3.1 基本流程

C 均值聚类算法串行程序的并行化设计可分为以下几个阶段:首先用 IPA 高性能工具得到热点函数的花费时间和并行情况,分析串行程序的可并行性[5];然后运用 TBB 和 OpenMP 进行并行优化设计;最后使用 IPA 的 Compare Results 功能进行比较,测试并行程序的性能效果。基本流程如图 1 所示。

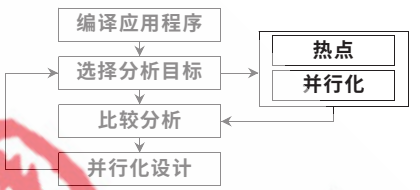

图 1 运行 Intel Parallel Amplifier 流程框架

3.2 热点定位

通过"Hotspot"可以获得热点函数所花费的时间,调用栈信息可以得到它被不同函数调用花费的时间。IPA 采集的数据为程序段花费的总时间、CPU 运行的时间、CPU 空闲时间、处理器的核数、执行程序的线程数等。找到热点函数后,打开源代码,分析哪些代码花费处理器时间最多。

3.3 并发性分析

Concurrency 分析可以得到热点函数在执行过程中各个其他任务并行执行的情况,以及各个线程的任务分配情况。IPA 并发性分析不仅包含热点采集的时间数据,更重要的是程序的并发状态。它用 5 种不同状态(Idle、Poor、Ok、Ideal、Over)表示并发性的情况。在多核平台下,理想的状态应该达到 Ok 以上,也就是说当热点函数运行时,其他线程同时工作在处理器上,这样可以提高多核资源的利用率。

3.4 串行程序优化

通过分析源代码,可以对串行程序进行如图 2 所示的并行优化。

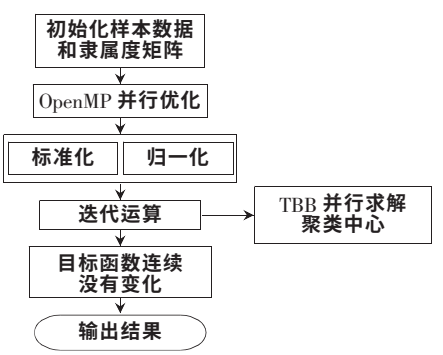

图 2 并行优化流程图

(1) 因为隶属度矩阵的归一化和样本矩阵的标准化没有数据相关性,所以可以利用 OpenMP 的工作分区功

能在两个线程中同时执行运算，提高多核的利用率，节省程序运行时间。使用 OpenMP 的优化设计：

```
#include <omp.h>
初始化数据
#pragma omp parallel sections//工作分区
{#pragma omp setion
  样本数据标准化
#pragma omp section
  隶属度矩阵归一化}
```

(2) 归一化后的隶属度矩阵和标准化的样本数据做矩阵乘法的运算，可以使用 TBB 并行库进行优化设计[6-7]。TBB::block_range2d 表示的是二维迭代空间的模板类，它包含在头文件 TBB/blocked_range.h 中，作用是根据需求对并行任务正确的划分。因为矩阵相乘是二维空间的运算，因此采用 block_range2d 模板类。迭代空间划分好后，就可以使用 TBB::parallel_for 执行并行操作。parallel_for 包含在头文件 TBB/parallel_for.h 中，作用是对循环体进行并行化处理。使用 TBB 的优化设计：

```
#include "tbb/taske_scheduler_init.h"
#include "tbb/parallel_for.h"
#include "tbb/blocked_range2d.h"
task_scheduler_init init;                    //初始化对象
{//矩阵相乘的 tbb 并行化
parallelMul ()double c, double a,double b}{parallel _for
(blocked_range2d<size_t>(0,k,0,n),MatrixlMul(c,a,b));}
}
```

4 实验结果测试

本文采用 UCI 标准数据集中的 Wine 数据集作为测试实例，该数据集包含有 178 个样本，每个样本有 13 个属性特征，分为 3 类，每类分别为 59，71，48，数据为 178×13 的矩阵。设定加权指数 m=2，停止阈值 ω=le-4。

(1)实验平台

硬件：Intel Pentium Dual T3400 @2.16 GHz 2.16 GHz，2 GB 内存。

软件：Microsoft Windows XP professional service pack3 操作系统；Visual.Studio.2008 英文专业版；parallel_studio_sp1_setup(评估版)；tbb22_009oss_win(TBB2.2 版本)[8]。

为了检测并行优化的效果，要对测试结果、热点、并发性和串行程序进行对比。

(2)实验结果

经过实验测试获得 Wine 数据集 3 个分类的样本数，分别为 59、64、48，与标准分类相比误差很小。本文通过 5 次运行 FMC 得到的实验结果相同，说明模糊 C 均值算法的并行优化设计是可行的。

(3)热点对比

从图 3 可以看到并行后热点函数 Update 执行时间减少为 15.321 ms，这是由于 Update 函数中有二维矩阵的并行化设计。在双核平台下，串行程序的线程数为 1，而并行程序的线程数为 3。

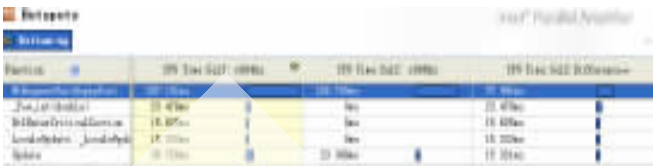

图 3 热点时间对比图

表 1 是 IPA 中 Compare Results 功能的比较结果，各项时间的差值都为正数，表明性能提高。

表 1 时间性能比较

性能比较	程序执行时间/s	CPU 运算时间/s	CPU 等待时间/s
串行程序	0.928	0.41	0.723
并行程序	0.884	0.39	0.649
时间差	0.084	0.02	0.074

(4)并发性对比

从图 4 可以看到并行程序的并发效果。热点函数 Update 并行后不仅时间减少了，状态也由 Poor 变为 Ideal。说明当热点函数运行时，其他线程同时运行在多核处理器上，多核利用率得到提高。

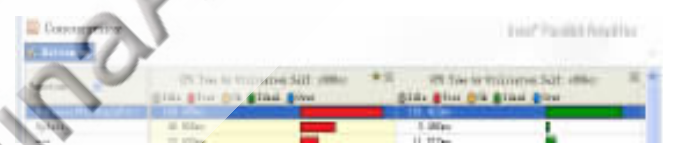

图 4 串行和并行程序并发性对比

本文将 Intel 多核高性能工具应用到 FMC 串行程序的并行优化设计中，提出并行优化设计方案，把 TBB 和 OpenMP 引入到聚类算法的并行化设计中。并行聚类算法在处理海量数据时将大大节省时间，并且提高多核资源的利用率。下一步的工作是从并行算法的可扩展性进行探究，并在众核处理器上做进一步测试，以便更好地提高聚类算法效率。

参考文献

[1] 齐淼，张化祥.改进的模糊 c 均值聚类算法研究[J].计算机工程与应用，2009，45(20)：133-135.
[2] 英特尔®软件网络[EB/OL].http://software.intel.com/en-us/intel-parallel-studio-home.
[3] REINDERS J. Intel threading building blocks [M]. O'REILLY 出版社，2007.
[4] 周伟明.多核计算与程序设计[M].武汉：华中科技大学出版社，2009.

[5] Peter Wang.使用 Intel parallel Amplifier:一站式解决最佳方案[EB/OL]. http://software.intel.com/zh-cn/blogs2010-2-22.

[6] 曹婷婷.基于多核处理器串行程序并行化改造和性能分析[D].成都:西南交通大学,2009.

[7] 胡斌, 袁道华.TBB 多核编程及其混合编程模型的研究[J].计算机技术与发展,2009,19(2):89-101.

[8] Intel 公司. Intel threading building blocks reference manual [EB/OL]. 2007. http: //threadingbuildingblocks. org/.

(收稿日期:2010-04-01)

作者简介:

张建强,男,1981 年生,硕士研究生,主要研究方向:并行计算、多核计算机系统。

郑晓薇,女,1957 年生,教授,主要研究方向:并行计算、多核计算机系统。

吴华平,男,1984 年生,硕士研究生,主要研究方向:并行计算、多核计算机系统。