

文章编号:1673-0062(2018)01-0073-05

基于多核系统的隐马尔可夫模型并行算法研究

龚向坚, 邹腊梅

(南华大学 计算机学院 湖南 衡阳 421001)

摘 要:介绍了一种基于 OpenMP 的多核并行程序设计方法,并使用此方法实现了对基于隐马尔可夫模型的 Web 文本挖掘程序的并行化改造.实验证明重新设计的多核并行程序相比于原有串行程序在多核微机系统上的运行时间大大减少、程序整体性能得到明显提升.

关键词:多核系统;并行计算;隐马尔可夫模型;数据挖掘

中图分类号:TP391.1 **文献标志码:**B

The Research on Parallel Algorithm of Hidden Markov Model Based on Multi-core System

GONG Xiang-jian, ZOU La-mei

(School of Computer, University of South China, Hengyang, Hunan 421001, China)

Abstract: This paper introduces a multi-core parallel programs design method based on OpenMP, and used this method to implement the parallel reconstruction program of Web text mining based on Hidden Markov Model. The experiments show that the redesign of the multi-core parallel program compared to the original serial program greatly reduce the running time of the program, the overall program performance has been significantly improved.

key words: multi-core system; parallel computing; Hidden Markov Model; data mining

0 引 言

随着多核处理器^[1]的广泛应用和网络数据量的迅速增加,传统串行程序既无法充分发挥多核处理器的硬件性能,也不能满足数据量迅速增加带来的对计算处理能力不断提升的需求,为解

决上述两个问题将传统微机系统串行程序改造为多核并行程序是一个行之有效的方法,也是必由之路.本文将基于隐马尔可夫模型的 Web 文本挖掘为例来探讨如何对传统串行程序进行多核并行化改造以解决上述存在的两个问题.

收稿日期:2018-01-05

基金项目:湖南省教育厅科学研究资助项目(15C1185)

作者简介:龚向坚(1977-),男,讲师,硕士,主要从事并行计算、软件测试方面的研究.E-mail:59518705@qq.com.

1 基于 OpenMP 的多核并行程序设计方法

多核处理器目前被广泛使用于微机系统,而传统的微机系统应用程序大多是串行程序,要想在多核微机系统中提高这些程序的性能,并充分发挥多核处理器的性能,必须仔细地对这些程序进行多核并行化^[2]改造,而如何挖掘这些程序的并行性、编写并行程序,并在多核微机系统上高效地运行,成为一个迫切需要解决的技术问题。

OpenMP 是一种面向共享存储器的多处理器多线程并行编程语言^[3],主要是为共享存储计算机上的并行程序设计使用的,而多核系统正是多个处理器核心之间共享主存储器的并行处理系统;OpenMP 并行编程语言提供了一组与计算机硬件平台无关的编译命令和函数,可以显式地指导编译器如何以及何时利用应用程序中的并行性。OpenMP 通过对原有的串行代码插入一些指导性的编译命令,并进行一些必要的修改,就可以快速地实现多核系统上的并行编程。因此使用 OpenMP 来进行多核并行程序设计^[4]既简单易行又能够有效地缩短程序运行时间、提高程序的整体运行性能。

2 隐马尔可夫模型及其在 Web 文本挖掘中的应用

隐马尔可夫模型 (Hidden Markov models, HMM)^[5-6]是一个双重随机过程,具体的状态序列不可知,只知其状态转移的概率,即模型的状态转换过程是不可观察的(隐蔽的),而可观察的事件的随机过程是隐蔽的状态转换过程的随机函数。

利用 HMM 进行 Web 文本挖掘^[7-8]因易于建立、适应性强、挖掘精度高的优点而日益受到关注,要将 HMM 应用于 Web 文本挖掘过程中,需要解决三个问题:(1)在给定观察序列 $O = O_1, O_2, \dots, O_T$ 条件下,调整 HMM 模型的各个参数,使得概率 $P(O|\lambda)$ 最大化;(2)在给定观察序列 $O = O_1, O_2, \dots, O_T$, 与模型 λ 的前提下,如何求 $P(O|\lambda)$ 值;(3)在给定观察序列 $O = O_1, O_2, \dots, O_T$, 与模型 λ 的前提下,如何求出状态序列 $S = q_1, q_2, \dots, q_T$, 使得 S 出现的可能性最大。问题(1)可结合遗传算法与 Baum-Welch 算法解决;问题(2)可利用前向和后向算法解决;问题(3)可用 Viterbi 搜索算法解决。

因此我们可将基于 HMM 的 Web 文本挖掘过程^[9]描述为:首先在确定 HMM 初始参数的过程中运行遗传算法,优化 HMM 参数;然后利用 Baum-Welch 算法进行 HMM 的训练^[10],构建隐马尔可夫模型;最后利用 Viterbi 算法进行测试样本最佳状态序列的求解。

3 基于隐马尔可夫模型的 Web 文本挖掘算法并行化设计

3.1 基于隐马尔可夫模型的 Web 文本挖掘串行算法分析

3.1.1 程序热点

程序热点是指程序运行周期中运行时间最长的部分,我们对基于隐马尔可夫模型的 Web 文本挖掘串行算法进行并行化改造关键的问题就是找出原有程序的程序热点然后对其进行并行化改造以实现执行效率的提升。

通过对基于隐马尔可夫模型的 Web 文本挖掘传统串行程序的分析我们可以知道整个算法包含三个主要部分:HMM 参数优化(遗传算法)、HMM 训练(Baum-Welch 算法)和 HMM 解码(Viterbi 算法)。经过对程序反复的运行测试发现 HMM 参数优化(遗传算法)在整个程序运行中时间占比非常大、地位非常稳定,因此,可以判定 HMM 参数优化(遗传算法)为程序的主要热点,要想通过并行化改造程序,以提高算法的执行效率,就必须从 HMM 参数优化(遗传算法)着手,提高这个部分的执行效率,减少执行时间,从而有利于整个算法整体性能的提升。

以 HMM 参数优化为中心,重点研究讨论如何使用基于 OpenMP 的多核并行程序设计方法对遗传算法进行并行化改造设计以提升对基于隐马尔可夫模型的 Web 文本挖掘程序整体运行效率和性能。

3.1.2 遗传算法分析

遗传算法流程图如图 1 所示。遗传算法的基本过程^[11-12]可描述为:1)随机产生 M 个初始个体,组成第一代种群;2)计算种群中每个个体的适应度,适应度函数的选择会影响寻优的结果,为了保证结果的最优化,我们选择采用 $P(O|\lambda)$ 作为适应度函数;3)根据适应度从上一代种群中选择 M 个个体复制到新一代种群中,我们把这一部分命名为选择算子模块;4)在新种群中选择两个个体进行交叉操作,重复 $M/2$ 次,产生 M 个新个体也放入新一代种群,我们把这一部分命名为交

叉算子模块;5) 选择变异点, 依据变异概率进行变异操作, 重复 M 次, 同样产生 M 个新个体放入新一代种群, 我们把这一部分命名为变异算子模块;6) 判断终止条件是否满足, 如果满足, 算法终止运行, 否则重复第 2 步至第 5 步, 遗传算法终止条件可以有两个: 达到最大迭代次数或达到规定

最小偏差. 遗传算法是个重复迭代的过程, 显然选择算子模块、交叉算子模块、变异算子模块这三个部分(图 1 中三个虚线框内部分)将会大量重复运行, 而且种群规模越大其运行在整个程序中所占的时间比例将会越大, 也自然会成为我们进行程序并行化改造、提高程序性能关注的焦点.

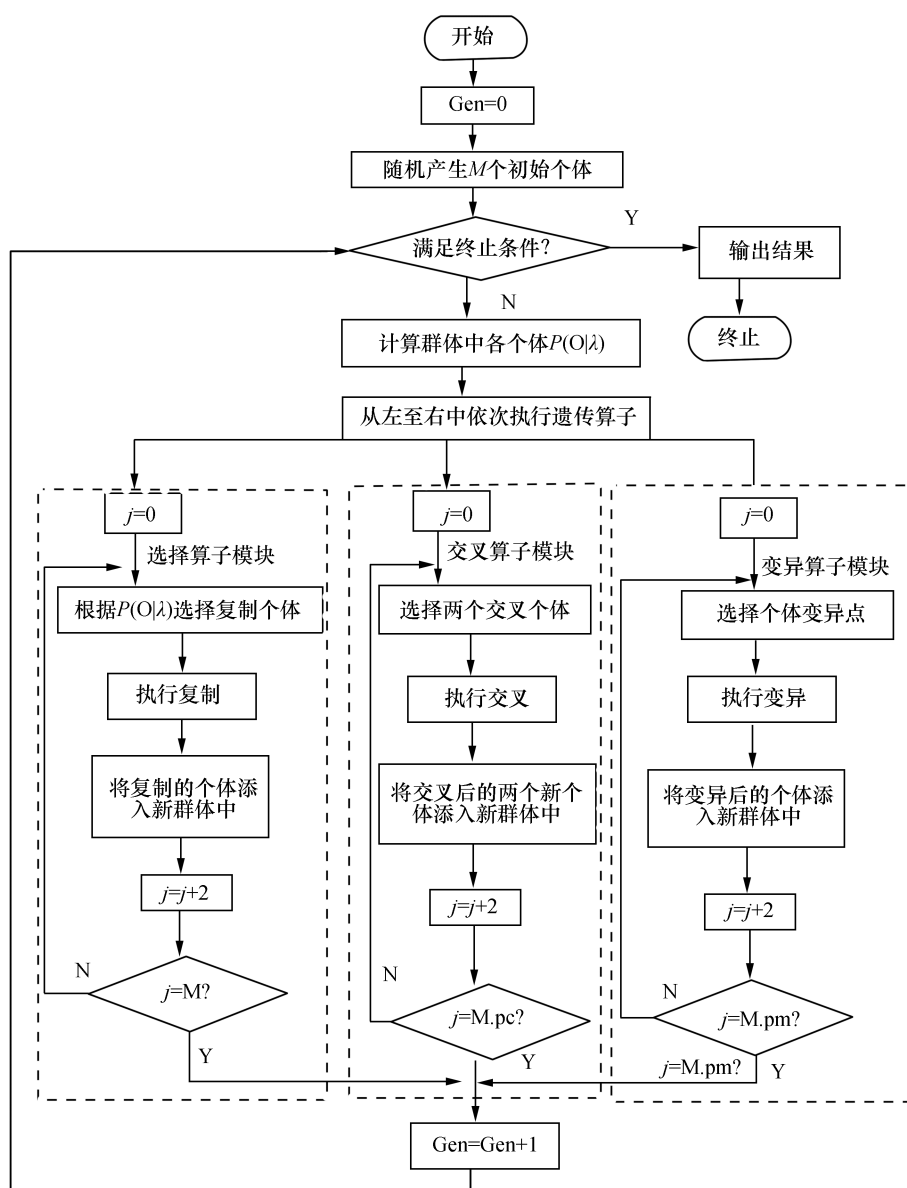


图 1 遗传算法的流程图

Fig.1 The flow chart of genetic algorithm

遗传算法执行过程中有 4 个重要运行参数: 种群大小 (M), 即群体中所含个体的数量, 一般取为 20 ~ 200; 最大迭代次数 (T), 一般取为 10 ~ 100; 交叉概率 (P_c), 一般取为 0.1 ~ 0.9; 变异概率 (P_m), 一般取为 0.01 ~ 0.1. 其中参数种群大小理论上而言越大越好, 因为这样遗传操作所处理的

模式就较多, 求得最优解的概率就越大, 但是随着种群规模增大, 遗传算法的计算量也会随之增加, 算法时间和空间的支出也会大幅增加, 会大量降低程序的运行效率, 因此在串行算法中一般只能取 20 ~ 200, 而如果对相应程序采用并行算法设计然后在多核系统上运行则可以在同等执行时间条

件下大大增加每个种群中所含个体的数量,从而提升整个算法的性能和效率,在多核并行算法中我们尝试将种群大小最大值取到 1 000,实验结果证明规模扩大之后程序结果优化明显,而程序运行所需的时间却并没有明显增加,种群规模为 1 000 的并行程序和种群规模为 200 的串行程序两者运行时间却相差不多.

3.2 基于隐马尔可夫模型的 Web 文本挖掘并行程序设计

通过上述对遗传算法串行政程序的分析可知遗传算法的程序热点是选择算子模块、交叉算子模

块和变异算子模块(图 1 中三个虚线框内部分),遗传算法的并行化改造主要任务就是要提升选择算子模块、交叉算子模块和变异算子模块的运行性能,因此,对遗传算法并行化改造,核心就是探索使用并行设计方法对选择算子模块、交叉算子模块和变异算子模块进行并行化改造.

通过对遗传算法可并行性分析可知,遗传算法并行模型可以从种群并行这个角度进行考虑,我们先分析建立如图 2 所示的遗传算法种群并行模型.

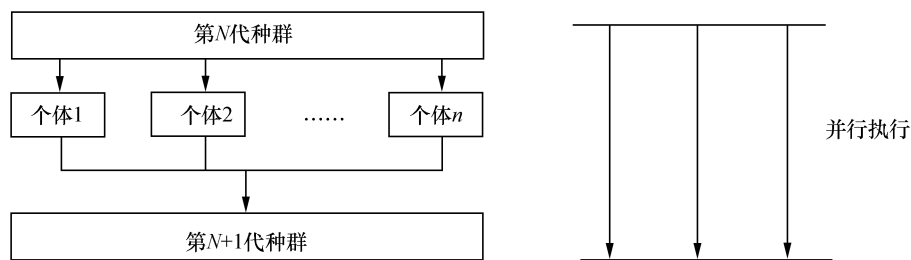


图 2 种群并行模型

Fig.2 Population parallel model

使用上述种群并行模型对遗传算法的程序热点选择算子模块、交叉算子模块和变异算子模块进行并行化改造设计,并使用基于 OpenMP 的多核并行程序设计方法进行并行程序设计,然后在多核实验平台上运行以测试其性能并进行相应分析和优化.

4 实验测试与结果分析

开发、测试运行平台为:开发环境使用 Visual

C++ 8.0(其对 OpenMP 有非常良好的支持),运行环境为 core i5-2300 2.8 GHz 四核 CPU 和 windows 7 操作系统.在实验过程中,针对遗传算法的主要参数我们选择设定了多个不同的测试用例,以观察不同种群规模下程序的运行时间以及加速比和不同种群规模对 HMM 参数最优解的影响.

遗传算法中种群规模扩大对 HMM 初始参数优化效果的影响,实验结果如表 1 所示.

表 1 遗传算法优化 HMM 参数

Table 1 Genetic algorithm optimization of HMM parameters

种群大小(M)	交叉率(P_c)	变异率(P_m)	种群中最好 $P(O \lambda)$	种群中最差 $P(O \lambda)$	种群平均 $P(O \lambda)$
50	0.1	0.01	7.659 864	4.656 923	6.873 869
200	0.1	0.01	8.198 756	4.639 187	7.589 934
500	0.1	0.01	8.586 219	4.718 519	7.926 418
1 000	0.1	0.01	9.570 936	4.839 589	8.634 247

通过分析实验结果我们发现,随着种群的增大,种群中最好个体的 $P(O|\lambda)$ 也在增大,证明了当 M 值较大时,遗传算法搜索范围扩大,求得最优个体的机会就随之增加.尤其当种群 M 增大到 1 000, $P(O|\lambda)$ 达到 9.570 936,种群平均

$P(O|\lambda)$ 达到 8.634 247,充分证明了遗传算法在大范围内寻优的特性,如果有足够的计算能力,遗传算法处理的种群规模越大其寻优能力越强.因此我们对传统应用程序进行多核并行化改造以提高他们的数据处理能力从而优化程序的性能是完

全有效可行的.

改造后的并行化程序与传统串程序的运行

时间及加速比,四个不同测试用例实验对比结果如表 2 所示.

表 2 运行时间和加速比

Table 2 Running time and acceleration ratio

测试用例	种群规模	串行算法运行时间/s	并行算法运行时间/s	加速比
HMMTest01	50	11.43	6.74	1.70
HMMTest02	200	31.22	15.07	2.07
HMMTest03	500	52.60	21.80	2.41
HMMTest04	1 000	95.31	32.12	2.97

实验结果表明:并行算法执行效率改善明显,执行时间相比串行算法有较大减少,并且随着遗传算法中种群规模的扩大其计算量增加运行时间也增加,相应的加速比也会逐步增加,也就是说随着规模扩大计算量增加并行化设计带来的性能改善更加明显.

5 结 论

通过实验数据的对比分析,我们发现基于多核系统的并行算法不仅能直接减少程序的执行时间改善程序的执行效率,同时可以通过多核并行程序设计来充分发挥多核处理器的性能,以扩大可计算处理的种群规模来提高 HMM 参数优化的结果,从而提升相应程序的整体性能.因此我们通过多核并行程序设计既可以充分发挥多核微机系统的硬件性能又可以优化应用程序的性能以提升其处理更大规模数据量和更复杂计算的能力.

参考文献:

- [1] OLUKOTUN K, NAYFEH B A, HAMMOND L, et al. The case for a single-chip multiprocessor[J]. Acm sigplan notices, 1996, 31(9): 2-11.
- [2] 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.
- [3] 游佐勇. OpenMP 并行编程模型与性能优化方法的研究

及应用[D]. 成都: 成都理工大学, 2011.

- [4] 龚向坚. 基于 OpenMP 的多核系统并行程序设计方法研究[J]. 南华大学学报(自然科学版), 2013, 27(1): 64-68.
- [5] 侯传宇. 马尔可夫及隐马尔可夫模型在数据挖掘中的应用[J]. 电脑知识与技术, 2008(3): 1186-1188.
- [6] 谢锦辉. 隐 Markov 模型(HMM)及其在语音处理中的应用[M]. 武汉: 华中理工大学出版社, 1995.
- [7] 王继成, 潘金贵, 张福炎. Web 文本挖掘技术研究[J]. 计算机研究与发展, 2000, 37(5): 513-520.
- [8] HU X T, HUANG Y A, YIN Z P, et al. Driving force planning in shield tunneling based on Markov decision processes[J]. Science China technological sciences, 2012, 55(4): 68-73.
- [9] 邹腊梅. 基于隐马尔可夫模型的 Web 文本挖掘技术研究[D]. 衡阳: 南华大学, 2007.
- [10] LIU J H, CHENG J S, CHEN J P. Optimization of HMM parameters based on chaos and genetic algorithm for hand gesture recognition[J]. Journal of systems engineering and electronics, 2002, 13(4): 79-84.
- [11] KWONG S, CHAN C W, MAN K F, et al. Optimization of HMM topology and its model parameters by genetic algorithms[J]. Pattern recognition, 2001, 34(2): 509-522.
- [12] ASLLANI A, LARI A. Using genetic algorithm for dynamic and multiple criteria website optimizations[J]. European journal of operational research, 2007, 176(3): 1767-1777.

(责任编辑: 龙威)