文章编号：1672-5913(2012)14-0073-05　　中图分类号　G642

并行程序设计课程学生作业的自动评判方法

赵长海，晏海华，贾宝龙，王艳雷，高　非
（北京航空航天大学 计算机学院，北京 100191）

摘　要：针对 SMP 架构的多线程并行程序和集群架构的 MPI 并行程序，提出从正确性、性能和可扩展性三个方面进行自动评判的方法，同时还提出对正确性、性能和可扩展性三个指标进行量化的方法，并基于量化值对并行程序自动判分。

关键词：并行程序设计；并行计算；多核；虚拟机；自动评判；自动评测

随着多核体系结构的出现和发展，使得并行计算科学的硬件基础设施发生了很大的变化，即使廉价的个人计算机也有多个处理器可供使用，只有掌握并行程序设计技术，才能有效利用计算资源。作为计算机人才培养主要基地的高等院校来说，配合该学科发展的课程建设刻不容缓。

北京航空航天大学计算机学院在大三下学期开设了并行程序设计课程，授课目标是帮助学生融会贯通先前所学的计算机系统基础知识、理解并行程序的运行机制、掌握并行程序设计方法。授课内容主要包括：单机的多进程编程、MPI 程序设计、并行程序性能分析，简单介绍并行算法设计与并行模式。

并行程序设计是实践性非常强的课程，必须重视学生动手实验，通过编程作业锻炼学生的并行程序设计能力，我院每年选修该课程的人数超过 130 人，人工评判学生作业将耗费巨大的工作量，非常不现实。相比串行程序，并行程序不仅仅要检查正确性，还要评判性能、可扩展性、效率等指标，这些指标必须基于大量的测试数据、在多台不同配置的机器上多次执行程序、对运行时信息统计分析后才能得到；根据获得的运行时数据，如何自动判断程序的优劣并判分，当前也没有相关的研究；如何检查作业抄袭，督促学生独立完成作业，也是一个难题。

为解决上述问题，我们在 CourseGrading[1] 的基础上进行了功能扩充，使其支持并行程序的自动评判与抄袭检测，经过一个学期的试用，获得了良好的教学效果。学生利用该系统可以及时得到详细的并行程序评测报告，并通过“性能、扩展性与得分排行榜”功能激发了学习兴趣和积极性；教师也从繁重的批改作业的任务中解放出来，腾出更多的时间通过系统的答疑论坛与学生进行交流；利用抄袭检测功能可以基本杜绝学生互相抄袭作业的现象。

1　相关研究

串行程序自动评判系统研究已经比较成熟，在高级语言程序设计教学中起到了举足轻重的作用。国内比较知名的系统有北航的 CourseGrading 系统（http://course.sei.buaa.edu.cn）、清华大学的 Tsinsen 系统（http://oj.tsinsen.com）、北京大学用于 ACM 竞赛练习的 Online Judge（http://acm.pku.edu.cn/JudgeOnline/），以及基于 Judge Online 开发的程序设计教学辅助系统 [2]；国外知名的有 ASSYST[3]、BOSS[4] 和 CourseMaster(现已改

基金项目：北京航空航天大学教改项目“并行程序设计教学辅助平台”(2010-35)。
作者简介：赵长海，男，讲师，研究方向为并行计算与系统软件；晏海华，男，副教授，研究方向为软件测试。

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

名为 CourseMarker)[5] 等。这类系统主要采用通过黑盒测试方法验证程序的正确性，即通过运行被测程序，检查运行结果与预期结果的差异，串行程序的正确性测试方法同样可以应用于并行程序，但是该方法需要进一步改进，以检测由于数据竞争、死锁等原因引起的并行程序正确性问题。

国内外很多高校已经在本科教育中开设了并行程序设计类的课程，相关的教改成果可参考文献 [6-11]，这些成果多是偏重教学方法，教学工具方面也仅限于教学网站，不涉及自动评判。

Intel 的 Parallel Universe Portal[12] 可以在线评测并行程序在多核 CPU 上的可扩展性，测评实现方法是在多核服务器上，安装多个虚拟机，分别虚拟出不同的 CPU 核数，然后在这些虚拟机上依次执行提交的程序，最后给出执行时间和加速比图表。该系统只能评测 Windows 平台下的多线程并行程序，无法测评 MPI 并行程序，而且不能自动给并行程序打分，目前 Intel 已经暂停该服务。

2 评判与抄袭检测流程

只有可编译执行的源程序才能被自动评判，评判与抄袭检测流程如图 1 所示，学生将源程序在线提交，系统首先编译源程序，如果编译不通过，则终止评判，返回编译错误信息；编译通过之后，根据并行程序的类别，采用不同的评判方法，若是 MPI 程序，则调度不同数量的同构机器进行动态测试，若是单机运行的多线程程序，则在不同核数的机器上依次执行动态测试；如果动态测试期间程序崩溃或者运行时间过长，则终止评判，返回运行时错误原因；最后根据运行时信息，分析程序的正确性、性能和可扩展性，自动生成评判报告并判分。

教师可以对学生提交的源程序进行抄袭检测，并将同一来源的代码聚类，抄袭检测结果只供教师参考，并不参与自动评分。本系统的抄袭检测方法详见文献 [13]。

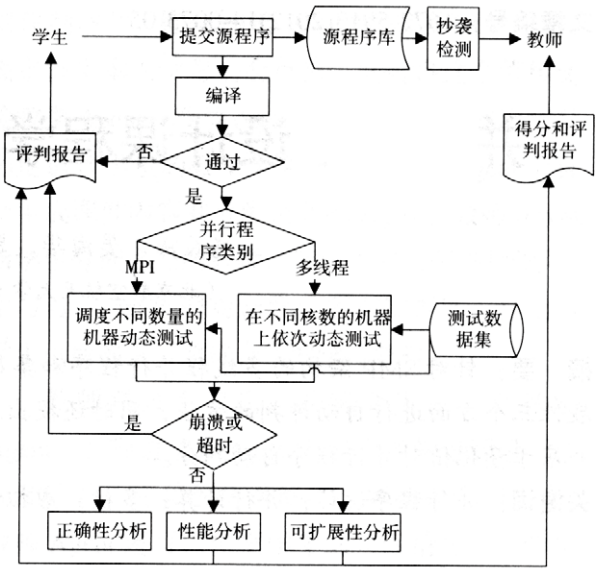

图 1 并行程序评判流程

3 自动评判方法

3.1 评判环境

根据访存模型，并行计算机主要分为共享内存和分布式内存两类。目前教学中，共享内存类主要讲授多核 SMP 架构上的多线程并行编程；分布式内存类，主要讲授集群架构上的 MPI 并行编程。

多线程并行程序的评判环境如图 2 所示，在配备有 N 个 CPU 核的服务器上安装 N/2 个虚拟机，分别虚拟 1、2、4……N 个 CPU 核，核心评判程序是基于 C/S 结构的，位于 Web 服务器上客户端将学生上传的源程序，依次提交到这些虚拟机上执行，记录执行时间，根据执行时间计算加速比，从而可以衡量程序的性能与可扩展性。

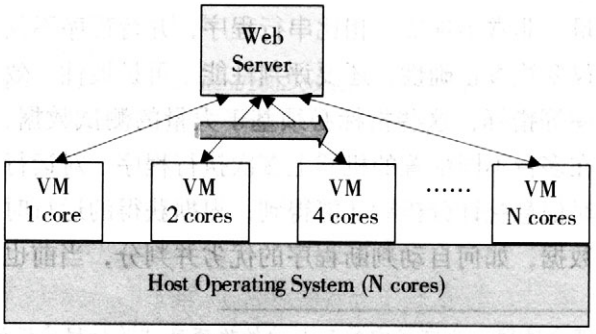

图 2 SMP 体系结构评判环境

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

MPI 并行程序的评判环境如图 3 所示，每台虚拟机的配置相同，所有虚拟机的 CPU 核数之和不超过底层主机的 CPU 核数，评判系统依次调度 1、2、3…n 台虚拟机执行被测程序。如果有一个真正的集群系统，直接使用物理机进行评判，结果会更加符合实际。

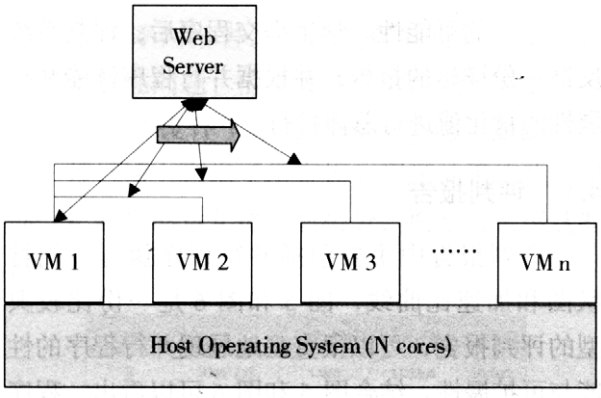

图 3　cluster 体系结构评判环境

3.2　正确性评判

本系统采用动态测试判断程序的正确性，所谓动态测试，是指通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率和稳定性等。程序的测试数据集由若干个测试用例组成，每一个测试用例包括输入数据与期望输出，一个测试用例对应一次程序的执行。任意一个测试用例不通过，则认为程序是不正确的，终止程序的评判。由数据竞争（data races）引发的并行程序正确性问题是最常见的，由于存在不确定性，可能会逃过系统的检查，适当增大测试数据量或者执行次数，可以降低逃过的概率。

我们将学生程序的运行时错误也归为正确性问题，它有两种表现形式：运行时间过长和崩溃。运行时间过长的主要原因是程序陷入了死循环，也可能是发生了死锁，另外一个原因是程序性能太差，不管是哪种原因必须及时终止程序运行，否则会增加服务器的负载，影响其他程序的评判。程序的最长运行时间由教师设定，超过这个时间，进程被强行终止。程序崩溃最常见的原因是对内存操作不当，例如数组越界、指针操作问题、堆栈溢出等。并行程序中数据竞争（或称为竞态条件）也会间接引起内存操作的问题。为方便学生查找崩溃原因，评判系统在被测程序内插装了信号处理函数，在运行时捕捉崩溃进程发出的信号，如 SIGFPE、SIGSEGV 和 SIGXFSZ 等，评判系统根据这些信号，列出引起程序崩溃的可能原因。

并行程序的正确性是最重要的，如果在所有的虚拟机上都测试正确，则会得到该题目 50% 的分数，否则得 0 分，并放弃后续的性能与可扩展性分析。

3.3　性能评判

如上所述，可以看出，假设有 n 台测试虚拟机，被测程序就会执行 n 次所有的测试用例，假设第 i 次执行的时间为 t_i，则并行程序的平均执行时间 T 由公式得到。

$$T = \frac{\sum_{i=1}^{n} t_i}{n}$$

T 值越小说明程序性能越高，我们分配题目总分数的 25% 用于奖励性能，用 S_P 表示性能总分数，所有提交的程序中，性能最高者（假设执行时间为 T_{min}）得到满分 S_p，执行时间为 T_i 的程序得分按公式计算得到。

$$SP \times \frac{T_{\min}}{T_i}$$

3.4　可扩展性评判

本文用加速比度量并行程序的可扩展性，多线程并行程序以单核虚拟机上的执行时间为基准计算加速比，MPI 并行程序以一台机器的执行时间为基准。加速比越接近线性增长，说明并行程序的可扩展性越好，例如图 4 中 L1 的可扩展性显然好于 L2。

为了方便对并行程序的可扩展性打分，本文提出一种量化并行程序可扩展性的方法：假设评判过程中最多使用了 N 个 CPU 核或者节点，用 *Alinear* 表示通过点（N, 1）的水平线和垂直线与线性加速比曲线围成的三角形面积，如图 4 中的阴影部分，用 A 表示通过该点的水平线和垂直线

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

与并行程序实际加速比曲线围成的多边形面积，那么并行程序的可扩展性量化值 *Scale* 等于 A 与 *Alinear* 的比值，值越大说明并行程序的可扩展性越好。这个比值很容易计算得到，假设评判并行程序的过程中依次调度 $P_1, P_2 \cdots\cdots P_n$ 个 CPU 核或者计算节点，对应的加速比分别为 $S_1, S_2 \cdots\cdots S_n$，那么 *Scale* 由公式计算得到。

$$Scale = \frac{\sum_{i=2}^{n} \frac{S_i + S_{i-1} - 2}{P_i - P_{i-1}}}{P_n^2}$$

一般情况下，*Scale* 值在 0 和 1 之间，有两类特殊情况：一类是超线性加速比，*Scale* 值可能大于 1，如图 4 中的超线性加速比曲线；另一类是并行程序没有扩展性的情况，如图 4 中的 L2 加速比曲线，根据公式，*Scale* 值会小于 0。

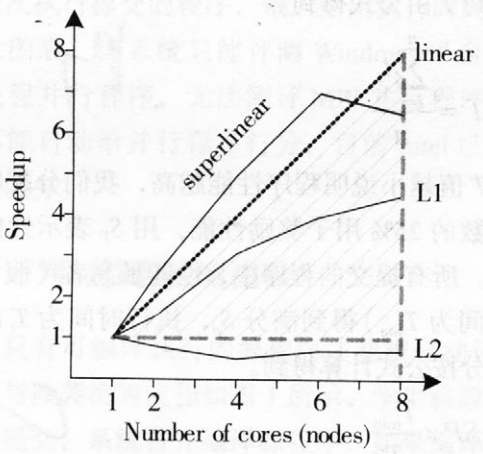

图 4 可扩展性量化方法

我们将题目总分数的 25% 用于并行程序的可扩展性，这部分分数用 *SC* 表示，*SC* 进一步分为 30% 的“基本分”和 70% 的“奖励分”，只要并行程序具备可扩展性，即 *Scale* 值大于 0，就能得到基本分。可扩展性自动判分规则如下：

若被测程序的 *Scale* 值小于等于 0，得 0 分；

若 *Scale* 值大于等于 1，得满分 *SC*；

若 *Scale* 值大于 0 且小于 1，假设 $Scale_{max}$ 是所有提交的程序中最接近 1 的一个值，可扩展性值为 $Scale_i$ 的并行程序的分数用公式计算得到。

$$SC \times 0.3 + SC \times 0.7 \times \frac{Scale_i}{Scale_{\max}}$$

4 应用实例

本节以一个实际作业题目展示评判系统的应用效果。题目要求：设计一个暴力破解 MD5 密码的多线程程序，该程序从标准输入读入一个长度为 32 的密文，输出该密文对应的用户口令。已知用户密码长度为 6 至 8 位，仅由数字组成。

该题目属于易并行类的问题，而且存在超线性加速比的可能性。学生提交程序后，评判系统反馈一份详尽的报告，并根据并行程序性能和扩展性的量化值进行总体排行。

4.1 评判报告

评判报告中主要包括并行程序执行时间柱状图和加速比曲线，图 5 和图 6 是一份比较典型的评判报告，可以很直观地展现并行程序的性能与可扩展性。结合图 5 和图 6 可以看出，程序使用两个核执行时，加速情况较为正常，而用到 4 个核时执行时间反而增加，加速比急剧下降至 0.886。从该现象并结合具体的问题，可以推断

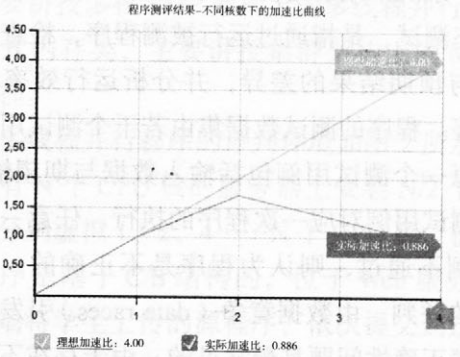

图 5 加速比曲线

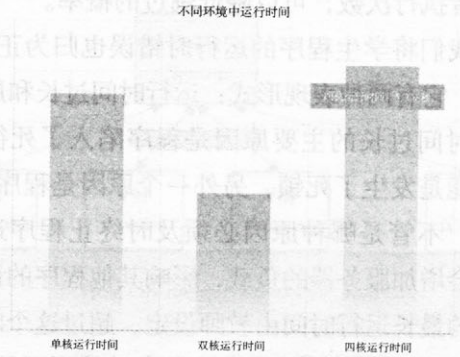

图 6 执行时间随 CPU 核数的变化情况

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

程序可能存在如下问题：并行任务间存在重复的计算；并行粒度太小，导致同步开销过大；计算任务分配不均匀导致处理器的负载不均衡。

学生利用该评判报告，可以直观地看到并行程序存在的问题，找到改进的方向，锻炼分析问题的能力，这是人工评判无法做到的。

4.2　排行榜

通过评判报告，可以发现并行程序是否存在可扩展性问题，但是无法确认性能的高低。为此，本系统设计了排行榜这一功能，如图 7 所示，排行榜上列出了所有提交程序的平均执行时间、扩展性的量化值及总得分，而且可根据这些值的大小进行排序。

作业题目	用户名	可扩展性	运行时间	分数
2	39061212	1.363	11.236s	20.000
2	38060118	0.920	11.198s	19.861
2	38060415	1.252	11.386s	19.780
2	38060201	0.946	11.750s	19.632
2	38060528	0.887	12.008s	19.312
2	38060322	0.890	13.086s	18.906
2	38060128	0.896	14.024s	18.807
2	38060318	0.878	13.794s	18.646
2	38060128	0.821	13.724s	18.479
2	38060609	0.527	14.158s	17.295

图 7　排行榜

排行榜起到了两方面的作用，一方面是学生通过对比发现自己程序的性能优化空间，弥补评判报告的不足；另外一方面是激励学生不断优化自己的程序，提升在排行榜上的名次和得分，从而加深他们对知识的理解和掌握。

5　结语

上机实践是并行程序设计课程的重要教学环节，但是评阅学生的作业面临巨大的工作量和挑战。本文针对 SMP 架构的多线程并行程序和集群架构的 MPI 并行程序，提出了从正确性、性能和可扩展性三个方面进行自动评判的方法。学生作业的评判结果以图表方式直观地展现，方便学生发现并行程序的问题并有针对性地改进，达到了人工评判无法企及的教学效果。本文还提出了对正确性、性能和可扩展性三个指标进行量化的方法，并基于量化值对并行程序自动判分，降低教师工作量。

GPGPU[14] 是并行计算领域的一个重要趋势，国内外已经有部分高校将 GPU 上 CUDA 并行编程引入课堂，我们计划在进一步的工作中研究和实现 CUDA 并行程序的自动评判方法。

参考文献：

[1] Zhao Changhai, Yan Haihua, Song Cong. A System for Automatic Assessment and Plagiarism Detection of Student Programs[C]. Proceedings of the International Conference on E-Business and E-Government, Guangzhou, 2010: 3619-3624.

[2] 罗英伟, 汪小林, 张正熠. 编程网格: 一个基于Online Judge的程序设计类课程教学辅助系统[J]. 计算机教育, 2009(2): 87-92.

[3] D Jackson, M Usher. Grading student programs using ASSYST[C]. SIGCSE'97, New York, 1997: 335-339.

[4] Joy M, Griffiths N, Boyatt R. The boss online submission and assessment system[J]. J Educ Resour Comput, 2005, 5(3): 2.

[5] C Higgins, P Symeonidis, A Tsintsifas. The marking system for CourseMaster[C]. Proceedings of the 7th annual conference on Innovation and technology in computer science education, Aarhus, Denmark, 2002: 46-50.

[6] T R Gross. Breadth in depth: a 1st year introduction to parallel programming[C]. SIGCSE'11, New York, 2011: 435-440.

[7] D Padua, U Vishkin, J C Carver.Joint UIUC/UMD Parallel Algorithms,Programming Course[C]. Proceedings of the NSF/TCPP Workshop on Parallel and Distributed Computing Education, in conj. with IPDPS. 2011.

[8] R Brown, E Shoop, J Adam, et al. Strategies for preparing computer science students for the multicore world[C]. ITiCSE-WGR'10, New York, 2010: 97-115.

[9] Rivoire S. A breadth-first course in multicore and manycore programming[C]. SIGCSE'10, New York, 2010: 214-218.

[10] 郭卫斌, 郑红, 李建华, 等. 并行与分布计算课程建设探索[J]. 计算机教育, 2009(3): 82-84.

[11] 陈国良, 孙广中, 徐云, 等. 并行计算课程的教学方法[J]. 中国大学教学, 2004(2): 35-37.

[12] Gaston Hillar. Cloud Computing: Detecting Scalability Problems[EB/OL]. [2011-12-16]. http://drdobbs.com/cpp/221900673.

[13] 赵长海, 晏海华, 金茂忠. 基于编译优化和反汇编的程序相似性检测方法[J]. 北京航空航天大学学报, 2008, 34(6): 711-715.

[14] Owens J D, Luebke D, Govindaraju N, et al. A Survey of General-Purpose Computation on Graphics Hardware[J]. Computer Graphics Forum. 2007, 26(1): 80-113.

（编辑：孙怡铭）

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net