基于 SimpleScalar 的多媒体应用程序特征分析①

李慧敏[1,2]，匡旺秋[3]，马英英[2]，谭 琰[2]

[1](湖南信息科学职业学院 计算机学院，长沙 410073)
[2](湖南大学 软件学院，长沙 410082)
[3](湖南师范大学 高师培训中心，长沙 410081)

摘 要：近些年来，多媒体产业正在以惊人的速度在发展。Intemet 的普及已经把多媒体技术推向了计算领域的主流地位。如何针对多媒体应用的程序特征，从计算机系统优化的角度有效提高计算机对多媒体的计算处理能力，成为了当今及未来几年内研究人员研究的热点方向。介绍了多媒体应用程序 MediaBench 及测试工具 SimpleScalar，提出一种程序特征分析工具集 PCA，最后进行程序测试，得出结果。

关键词：并行；多媒体应用程序；核心循环；程序特征

Program Loop Characteristics of Multimedia Applications Based on SimpleScalar

LI Hui-Min[1,2], KUANG Wang-Qiu[3], MA Ying-Ying[2], TAN Yan[2]

[1](Department of Computer Teaching, Hunan Information Science Vocational College, Changsha 410073, China)
[2](Department of Computer Teaching, Hunan University, Changsha 410081, China)
[3](Hunan Teachers Training Center For Higher Education Institutions, Hunan Normal University, Changsha 410081, China)

Abstract: In recent years, the multimedia industry has been growing at a tremendous rate. The proliferation of the Internet has Pushed multimedia into mainstream computing. It becomes a hot research direction to explore architecture techniques to meet the processing and computing demands for multimedia applications according to multimedia program characteristics and performance analysis. Introduces the MediaBench and SimpleScalar, design a new program analysis tool, PCA (program character analyzer). At last, test the Multimedia Applications and get a conclusion.

Key words: parallel; multimedia applications; loop characteristics; program characteristics

1 引言

多媒体信息的大量涌现要求现代计算机必然具备卓越的多媒体处理器能力，多媒体应用程序的一个最突出的特点就是其内在的并行性，要求同时对多个数据单元进行相同的规则操作，这种并行性被称为子字并行。为了有效挖掘和利用并行，当前大多数体系结构设计都对指令集进行了多媒体指令的扩展，以实现子字并行操作，但是，如何让编译器产生有效的并行代码却成为一个瓶颈。由于多媒体应用主要还是以串行程序方式编写，而结果却需要以并行的方式执行，因而对编译器提出了很高的要求。如何从普通的串行 C 程序中识别出子字并行指令，是一个前沿性的研究课题。而对多媒体应用程序特点的程序特征分析和研究，正是这些工作的基础和重点。

2 MediaBench简介

MediaBench 是一种典型的多媒体应用程序，是一套用 C 语言编写的源码公开的多媒体标准应用程序，它涵盖了图像、图形、视频、音频、语音等多个区域，包含大量编码解码程序，具有典型多媒体工作负载特征，具有很好的研究价值。表 1-1 列出了 MediaBench 中的 13 个代表性应用程序[1-3]。

3 SimpleScalar简介

SimpleScalar 是一个源代码公开的体系结构模拟器，是一个开放软件，最初设计的目的主要是用来模

① 基金项目:湖南师范大学青年科学基金(61005)
通讯作者：匡旺秋，email:hnkuang@126.com
收稿时间:2011-03-03;收到修改稿时间:2011-04-04

表 1　MediaBench 的 13 个代表性应用程序按类型进行分类

多媒体类型	应用程序包	应用程序	多媒体类型	应用程序包	应用程序
Image (图像)	EPIC	epic,unepic	Audio (音频)	ADPCM	rawcaudio,rawdaudio
	JPEG	cjpeg,djpeg	Speech (语音)	G.721	g721dec,g721enc
	Ghostscript	gs		GSM	gsmdecode,gsmencode
Video (视频)	H.263	h263dec,h263enc		RASTA	rasta
	MPEG-2	mpeg2dec,mpeg2enc	Security (安全)	Pegwit	pegwitdec,pegwitenc
	MPEG-4	mpeg4dec,mpeg4enc		PGP	pgpdecode,pgpencode
Graphics (图形)	Mesa	mipmap,osdemo, texgen			

拟处理器的，它的指令集使用了类 MIPS 指令集，还可以进行自定义和扩充。SimpleScalar 模拟器具有良好的可移植性和可扩展性，目前绝大多数类 Unix 平台都支持 SimpleScalar 模拟器，基于 windows 平台的模拟器移植工作已初步完成。而且模拟器能够支持各种不同层次设计人员的需求，模拟器自身内嵌了 6 个子模拟器(SimFast、Simsafe、SimProfile、SimCache、Sim-cheetah、SimOutorder)，从侧重执行速度的功能模拟到关心结构内部细节的性能模拟一应俱全[4]。选用 SimpleScalar 模拟器是因为其采用的是执行驱动的模拟方式，而不是基于 trace 的模拟[5]，可以更好的反映出被模拟程序实际运行时的情况，能够在不同体系结构模型下对应用程序进行一个非常详细的测试分析[6]。SimpleScalar 使我们能够把对 MediaBench 的程序特征分析与对计算机体系结构的探讨紧密的联系起来，极大的方便了我们研究工作的开展。而且 SimpleScalar 所提供的模拟环境和测试数据在国际上有着非常高的认知度和可信性[7]。为了满足测试的需要，我们还对 SimpleScalar 进行了修改。

4　程序特征分析工具集

传统的程序测试分析工具主要是从宏观角度针对程序整体进行特征提取的，但是如果想深入到程序内部，进一步了解程序局部或微观领域的特征时，这些现有的测试工具就表现出了很多不足之处。例如我们想进一步了解程序核心循环的特征时，尤其是核心循环的执行时间和循环次数、循环深度和宽度，一个核心循环的操作类型及操作数类型，读入写出的数据量，循环变量的递进规律，最内层循环体所进行的操作，循环内部有无 If 条件分支语句、数组、赋值(计算)语句，循环片之间的关系，数据相关性，多个循环之间的关系，单循环或组合循环的特点等。所以在针对核心循环和过程进行测试时，自行设计并实现了一个程序特征分析工具—PCA。

总的算法框架和流程图如下:

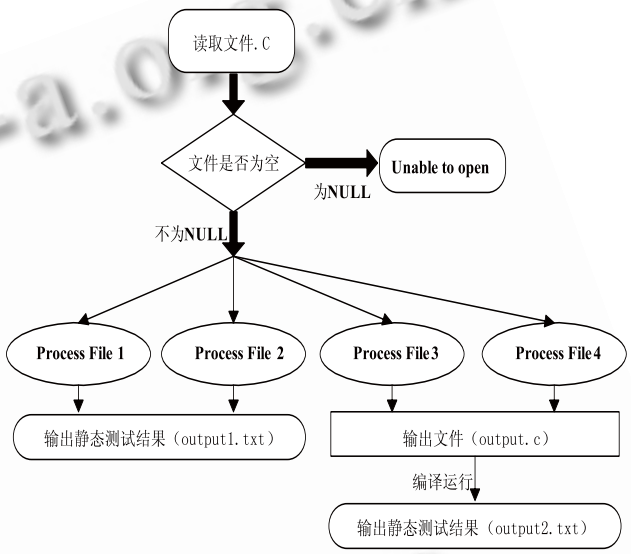

图 1　PCA 的主程序流程图

在图 1 中，函数 process_Filel 和 process_File2 的主要功能是完成程序的静态代码测试，主要针对程序中的 for 循环，并考虑了 for 循环形式的 2 种可能形式。根据对于 MediaBench 源程序的阅读和理解，总结了 MediaBench 内层循环的结构和特点。for 循环在结构上有两种可能，一种是循环的循环体在一堆匹配的大括号里;而另一种 for 循环的循环体仅仅只是一条语句，直接跟在 for 之后，前后都没有大括号。所以鉴于这种情况，编写了两个函数过程 process_Filel 和 process_File2，用来针对这两种不同情况查找和统计 for 循环。并且考虑了程序有注释的情况，在程序测试时跳过了对程序注释的统计。能够给出 MediaBench 中任意一段程序中的 for 循环数量、位置、循环体内容等数据。另外这两个函数过程还能够查询并统计 for 循环中的数组、if 条件分支、赋值语句的使用情况，并最终把结果保存在文件 output1.txt 里。函数 process_File3 和 process_File4 的主要功能是完成程序的动态测试。这两个函数执行后，生成了一个新的

output.c 文件。该文件经过重新编译并执行，可以得到被测程序的动态执行特征。如:程序中每个 for 循环的执行时间和循环被调用次数等动态数据。

程序实现关键技术及步骤（以查找赋值语句算法为例）:

1).查找“=”，如果找到，检查“=”前一个字符是否为*，/，%，+，-，&，|，^

如果是的话打印行号，打印出这个赋值操作。

2).如果“=”前的字符为“>”或“<”，则检查“>”或“<”前的字符是否为“>”或“<”，若是的话打印行号和这个赋值操作。

如果为单独的“=”，则打印行号及“=”操作。

如果以上三条满足任意一条，则打印“=”所在的这一行。

PCA 可以从程序静态和动态两个角度来分析和提取程序核心循环的特征。静态方面，能够给出 MediaBench 中任意一段程序中的 for 循环数量、位置、循环体内容等数据。另外还能够查询并统计 for 循环中的数组、if 条件分支、赋值语句的使用情况，并最终把结果保存在文件 outPutl.txt 里。动态方面，可以给出程序执行过程中每个 for 循环的执行时间和循环被调用次数等动态数据。PCA 的使用给多媒体程序的测试和分析工作，带来了很多便利之处，更给出了很多从前的程序测试工具所不能提供的测试分析数据。

5 程序测试

以在 Linuxredhat7.3 下(CPU:P4 2.8GHz，内存 512M)用 SimOutorder 测试 epic 编码程序为例。

1) 找到被测核心循环的开始点和结束点，在 for 循环的开始和结束点插入相应的标准系统调用(POSIX System Calls)语句 fopen. 这样就在核心循环 for 的开始和结束点打开了两个可写的文件 Psl1 和 Psl2。

2) 在 syscall.c 的文件头声明两个文件变量，，fopen 在执行过程中会在 syscall.c 中调用 case SS_SYS_open 过程，并且会在相应的 buf 中存有打开的文件的名字，如:Psl1。

3) 通过一个 if 比较匹配语句 if(strcmp (buf，”Psl1”)==0)把循环开始点的统计测试数据写入一个文件/home/Psl/myoutPut1。部分源代码如下:

```
循环开始点
/*my code loop start: open Psl1*/
if(strcmp(buf， ”Psll”)=0)/*compare strings label loop start*/
{
Printf(buf);/*Print filename*/
/*open a file—myoutput1 to save loop start stas*/
if((myoutputl=fopen(”/home/Psl/myoutputl”， ”w+”))==NULL)
{ fprintf(stderr ， ”Cannot Create FILE:/home/Psl/myoutputl”);
exit(1); }
/*Print start point simulation stats*/
fprintf(myoutputl， ”\nsim:**loop start simulation statistics**\n”);
stat_print_stats(sim_sdb，myoutputl);
sim_aux_stats(myoutputl);
fprintf(myoutputl，”\n”);
fclose(myoutput1);
```

4) 这样循环开始点和结束点 SimOutorder 的统计数据就分别写在了 /home/psl/myoutput1 与 /home/psl/myoutPut2 两个文件里，通过比较分析就可以得到 epic 编码程序核心循环的统计数据了。

表 2 EPIC 编码程序测试结果

Module (模块)	Function (函数)	Total Count (被调用总次数)	Time (运行时间)	Percent%(所占百分比)	Total Time(总运行时间，包含内部子过程)
Process	Thread_5a4		548	100	548
EPIC.EXE	Internal_filter	30	326	59.49	328
EPIC.EXE	Memset	1490	66	12.04	66
EPIC.EXE	Quantize_image	16	48	8.75	74
EPIC.EXE	_ftol	589910	25	4.56	25
Kernel32.dll	WriteFile	60	20	3.64	20
Kernel32.dll	ReadFile	141	16	2.92	16
EPIC.EXE	Run_length_encode_zeros	1	14	2.56	14
EPIC.EXE	Main	1	12	2.19	546

从表 2 可以看出，占用运行时间最长的是函数 Internal filter，占整个程序运行时间的 59. 49 %，函数 Internal filter 就是我们要找的核心函数。

6 结语

文章采用模拟器 SimpleScalar 对多媒体应用程序 EPIC 进行测试，发现 EPIC 编码程序在执行过程中,绝大部分的时间在运行函数 Internal filter，再利用前面提出的程序特征分析工具集 PCA 对 Internal_filter 进行测试分析，从结果可以得出，Internal filter 是在做内部滤波(卷积)，从 Internal filter 的源代码中可以看到,其核心循环是模板与原始图像九个部分分别求卷积，其中存在并行性。并行性可以通过循环展开进行优化，为其设计专门的体系结构自动调度循环执行，最终实现循环内部操作、表达式、循环体的多层次并行，加快程序执行速度，全面加速计算机对多媒体应用程序的计算与处理的速度，这些是我们下一步将要进行的工作。

参考文献

1 Austin TM. A User’s and Hacker’s Guide to the SimpleScalar Arehitectural Research Tool Set, 1997.

2 Bishop B, Kelliher TP, Irwin MJ. A Detailed Analysis of MediaBeneh. Proc. of the IEEE Workshop on Signal Processing Systems, 1999.

3 Bhargava R, John L, Evans B, Radhakrishnan R. Evaluating MMX technology using DSP and multimedia applications. Proc. of IEEE/ACM Sym. on Microarchit-Ecture, Dec. 1998: 37–46.

4 Ausrin T, Larson E, Ernst D. SimpleScalar:An in frastructure for computer system modeling. IEEE Computer, 2002, 35(2): 59–67.

5 彭绍亮,窦勇,李姗姗.面向多媒体应用的程序特征分析与研究.计算机工程与科学,2004,26(3).

6 陈剑龙,傅忠传,崔刚.SimpleScalar 模拟器内核分析及应用.哈尔滨工业大学学报,2004,36(5).

7 Brooks D, Tiwari V, Martonosi M. Wattch:A framework for architectural-evel Power analysis and optimizations. Proc. of 27th Ann Int’l Symp Computer Architecture. Los Alamitos: IEEE CS Press, 2000. 83–94.

（上接第 14 页）

10 周旋,王丽芳,蒋泽军.基于 Ajax 的即时消息系统的设计与实现.科学技术与工程,2009,9(2):446−450.

11 Chatterjee S, Abhichandani T, Li HQ. Instant Messaging and Presence Technologies for College Campuses. IEEE Network, 2005,19(3):4−13.

12 孙清国,朱玮,刘华军,张鹏.Web 应用中的服务器推送技术研究综述.计算机系统应用,2008,17(11):116−120.

13 Alexandre D, CalvaryKarin C. COMET(s), A Software Architecture Style and an Interactors Toolkit for Plastic User Interfaces. 15th International Workshop on Interactive Systems: Design, Specification, and Verification (DSV-IS 2008), Canada: Kingston, 2008.

14 王非.WebServices 应用研究与 RIA 系统中的实现.计算机应用与软件,2010,27(3):168−171.

15 曲海成,李洋.一种基于 Web 服务的 RIA 系统集成的应用.计算机系统应用,2010,19(9):15−18.