第35卷 第10期 计 算 机 工 程 2009年5月
Vol.35 No.10 Computer Engineering May 2009
·软件技术与数据库· 文章编号：1000—3428(2009)10—0030—03 文献标识码：A 中图分类号：TP311

并行计算模型参数动态分析软件包设计

王向前[1,2]，张云泉[1,3]，侯晓吻[4]
(1. 中国科学院软件研究所并行计算实验室，北京 100190；2. 中国科学院研究生院，北京 100190；
3. 中国科学院计算机科学国家重点实验室，北京 100190；4. 杭州电子科技大学，杭州 310018)

摘 要：并行计算模型的发展引入越来越多的模型参数。对并行计算模型参数动态采集分析软件包 DEMPAT 的整体框架进行研究，实现基于硬件性能计数器的存储层次参数采集模块。实验表明，该模块能够准确快速地获取存储层次参数且具有较好的可移植性。
关键词：并行计算模型；机器参数；存储层次

Design of Dynamic Analysis Toolkit for Parallel Computational Model Parameters

WANG Xiang-qian[1,2], ZHANG Yun-quan[1,3], HOU Xiao-wen[4]
(1. Lab of Parallel Computing, Institute of Software, Chinese Academy of Sciences, Beijing 100190;
2. Graduate University of Chinese Academy of Sciences, Beijing 100190;
3. State Key Lab of Computer Science, Chinese Academy of Sciences, Beijing 100190;
4. Hangzhou Dianzi University, Hangzhou 310018)

【Abstract】 More and more parameters are introduced with the development of parallel computational model. This paper carries out researches on the framework of a Dynamic Execution and Machine Parameter Analysis Toolkit(DEMPAT) for parallel computational model parameters. A portable memory hierarchy parameter acquisition module is implemented, which shows high speed and accuracy in the experiments.
【Key words】 parallel computational model; machine parameter; memory hierarchy

1 概述

从存储模型的角度把并行计算模型分为3代：共享存储(shared memory)并行计算模型，分布存储(distributed memory)并行计算模型和层次存储(hierarchical memory)并行计算模型[1]。并行计算模型从第1代发展到第3代的过程中引入了越来越多的模型参数，这在提高模型分析精度的同时也造成了更大的困难。

由于处理器新技术的飞速发展以及处理器与存储系统之间速度差的不断增大，把存储层次引入并行计算模型成为一种发展趋势。高速缓存和TLB(本文重点考虑数据高速缓存和数据 TLB)是存储层次中2个重要的组成部分，文献[2]中设计的高速缓存敏感(cache conscious)的算法能够使标准的查找算法获得2倍~5倍的性能提升，而自适应性能优化软件包和性能建模技术对获取精确的存储层次参数提出了更高的要求。但要获取高速缓存和TLB的参数并不是一件容易的事，因此，通过软件的方法来动态采集这些参数成为一个值得研究的课题。

2 软件包的整体框架

图1是本文提出的并行计算模型参数动态采集分析软件包 DEMPAT(Dynamic Execution and Machine Parameter Analysis Toolkit)的整体框架，主要由采集层和分析与表示层2部分组成。

为了全面获取硬件平台的特征和程序运行的细节，在采集层综合了静态与动态2种采集模块，分别获取硬件参数信息和程序运行时对硬件资源的利用情况。分析与表示层对采集层得到的结果进行分析和直观展示。

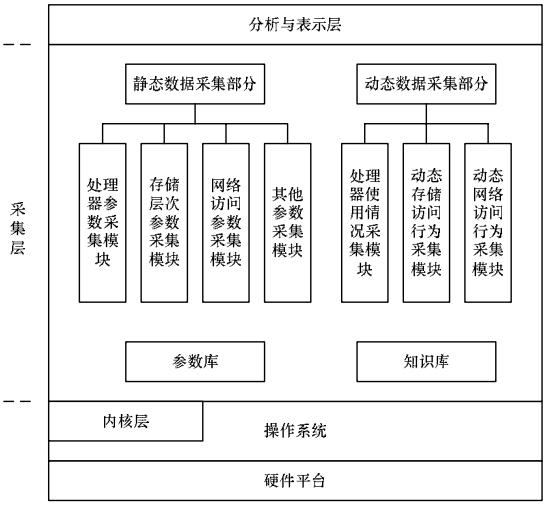

图1 软件包整体框架

基金项目：国家自然科学基金资助项目(60303020, 60533020)；国家“863”计划基金资助项目(2006AA01A102, 2006AA01A125)；北京邮电大学网络与交换技术国家重点实验室开放课题基金资助项目(2005-05)
作者简介：王向前(1982－)，男，硕士研究生，主研方向：并行性能测试与分析；张云泉，研究员、博士、博士生导师；侯晓吻，硕士研究生
收稿日期：2008-08-10 **E-mail：**jascha.wang@gmail.com

静态数据采集部分由以下 4 个执行模块组成：

(1)处理器参数采集模块：采集每个节点处理器的基本信息和处理能力，包括每个节点执行单元的数目(处理器数目及每个处理器中核的数目)、处理器的主频、处理器的计算能力(单位时间内整型和浮点型的运算次数)等。

(2)存储层次参数采集模块：存储层次参数是并行计算模型中新引入的评价指标，该模块搜集本地存储层次的结构参数和性能参数。结构参数包括：各级高速缓存的容量、行长(line size)和组相联(associativity)，各级 TLB 的容量、组相联，内存页面大小，主存容量和磁盘空间等；性能参数包括各级高速缓存的命中延迟(hit latency)和缺失损失(miss penalty)，主存的带宽和访问延迟以及磁盘的访问时间等。

(3)网络访问参数采集模块：网络访问参数是并行计算机系统尤其是分布式存储系统中非常重要的参数，该模块主要采集的参数有网络延迟、网络开销和网络带宽等。

(4)其他参数采集模块：采集其他参数指标，如 I/O 延迟等。

动态数据采集部分由以下 3 个执行模块组成：

(1)处理器使用情况采集模块：记录程序执行过程中处理器的运行情况，包括流水线的停顿、分支预测的执行情况和整型/浮点型指令操作数等。

(2)动态存储访问行为采集模块：获取程序执行过程中的访存行为，如存取操作指令条数、高速缓存访问和缺失、内存访问和页面缺失等。

(3)动态网络访问行为采集模块：记录程序中使用的通信模式和时间开销。表 1 给出了并行程序中用到的几种通信模式[3]。

表 1　并行程序中主要的通信模式

通信模式	对应的 MPI 函数
点对点通信	MPI_Send, MPI_Recv 等
集合通信一到多方式	MPI_Bcast, MPI_Scatter 等
集合通信多到一方式	MPI_Reduce, MPI_Gather 等
集合通信多到多方式	MPI_Alltoall, MPI_Allgather 等
远程存储访问 RMA	MPI_Get, MPI_Put 等

采集层除静态数据采集部分和动态数据采集部分外，还有 2 个辅助模块：参数库和知识库。参数库中存储常用硬件平台的参数信息，用户可以直接使用其中给定的硬件参数信息。虽然单个的并行程序执行细节各不相同，但某一类程序之间可能存在相似的计算和通信模式，文献[4]根据这种模式的相似性把科学计算领域的应用分成 7 个“小矮人”(dwarfs)。按照这种划分方法，在知识库中存放每一类程序的经验特征如计算和数据移动等，用来指导程序的动态数据采集。

3　2 种存储层次参数采集的方法

对给定的数组执行大量的访问操作时，高速缓存缺失(cache miss)的比率与数组的长度和相邻 2 次访问的步长存在一定的关系。假定高速缓存的容量为 C，行长为 B，组相联为 A，根据数组长度 N 和访问步长 S 的不同，缺失的比率可能出现如表 2 所示的 4 种情况(为方便讨论，假定 C 和 S 为 2 的幂次)[5]。基于时间的方法根据元素平均访问时间的不同来推断存储层次的各项参数。文献[5]提出了通过调整数组长度和访问步长来采集单层高速缓存和 TLB 参数的方法。文献[6]同时考虑了多层的存储结构并实现了 X-Ray。

表 2　高速缓存缺失情况

数组长度	步长	说明
$N \le C$	$1 \le S \le N/2$	经过强制缺失(compulsory miss)后缺失数为 0
$N > C$	$1 \le S \le B$	每 B/S 次元素访问会发生一次容量缺失(capacity miss)
$N > C$	$B < S \le N/A$	每次元素访问都会发生一次容量缺失
$N > C$	$N/A \le S \le N/2$	元素位于同一高速缓存组(set)中且没有冲突缺失(conflict miss)，缺失数为 0

硬件性能计数器也称为硬件计数器，是很多高性能处理器提供的一些特殊功能的寄存器，它用来跟踪底层硬件的操作和事件，并对相关活动进行计数。基于硬件计数器的方法根据访问数组时与高速缓存和 TLB 相关的硬件计数值的变化来推断其结构和性能参数，文献[7]提出了基于 PAPI 来采集高速缓存和 TLB 结构参数的方法。

基于时间的方法是一种比较通用的办法。但它不仅受到编译器对代码优化的影响，而且要事先对存储层次进行很多假定，一旦某些假定不成立就会严重影响结果的准确性。基于硬件计数器的方法能够很好地避免这些问题，而且多次运行同一程序代码段，高速缓存和 TLB 事件计数值很少会发生变化，因此，可以更快更准地获得存储结构信息。

4　基于 PAPI 的存储层次参数采集方法

4.1　模块整体框架

为了能够安全方便地访问硬件计数器，学者和研究机构着手开发了一系列通用的接口软件包，其中，PAPI (Performance Application Programming Interface)得到了硬件供应商、用户和性能工具开发者的一致认可，是目前适用平台最广泛的接口[8]。

存储层次参数采集模块框架如图 2 所示，虚线框中是几个主要部分：结构参数采集模块和性能参数采集模块分别负责采集存储层次的结构参数和性能参数，两者通过可移植的接口层 PAPI 访问底层硬件平台 数据分析模块对搜集到的数据进行查找分析，推断出存储层次的各项参数，结果交给上层的分析与表示层。

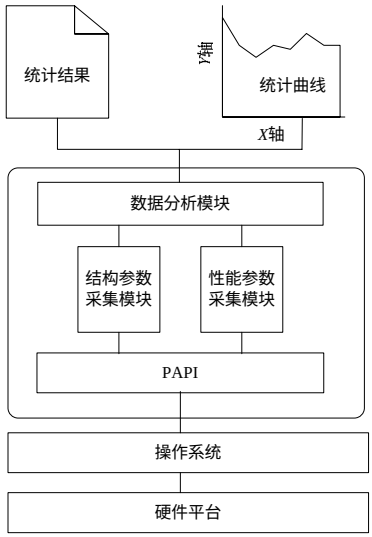

图 2　存储层次参数采集模块框架

4.2　结构参数采集模块的实现方法

首先实现存储层次中 2 个主要部分：高速缓存和 TLB 的结构参数采集。受硬件平台上可用预置计数事件的限制，待测计数事件与实测计数事件并不完全相同，表 3 列出了在 Pentium IV 平台上用到的 PAPI 预置计数事件。

表 3　选取的硬件计数器

待测计数事件	描述	实测计数事件
PAPI_L1_DCM	一级数据高速缓存的缺失次数	PAPI_L2_TCA-PAPI_L1_ICM
PAPI_L2_DCM	二级数据高速缓存的缺失次数	PAPI_L2_TCM
PAPI_L3_DCM	三级数据高速缓存的缺失次数	N/A
PAPI_TLB_DM	数据 TLB 的缺失次数	PAPI_TLB_DM

测量高速缓存容量时，固定访问步长为 1，同时增加数组的长度。当数组长度超过高速缓存容量时，缺失数会出现一次明显增加。

测量高速缓存行长和组相联时，固定数组长度(满足数组长度大于高速缓存容量)同时增加访问步长，当步长达到或超过行长时缺失数也会明显增加，在被访问数据映射到同一组后缺失数又降为 0。根据这几个突变点就可以推断高速缓存的结构参数，采集 TLB 参数的方法类似。

5　测试结果

在 Intel Pentium4 3.0 GHz 的处理器上采用整型数组进行了大量测试，结果如图 3 和图 4 所示。一级高速缓存的容量 C_1=16 KB，行长 $B_1=2^4\times4$ B=64 B，组相联 $A_1=(64\times2^{10})/(2^{11}\times4)=8$；TLB 表项对应的内存空间为 256 KB，内存页面大小为 $2^{10}\times4$ B =4 KB，TLB 的容量为 256/4=64。

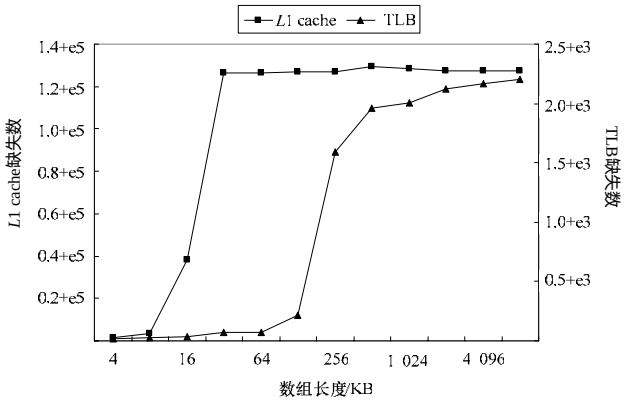

图 3　*L*1 cache 和 TLB 的大小

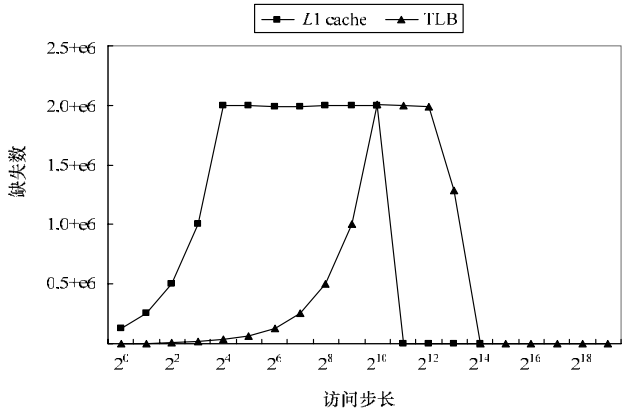

图 4　*L*1 cache 的行长、组相联和页面大小

通过与 Intel 处理器手册[9]和 CPU-Z 获取的结果进行对比，如表 4 所示，可见笔者的采集结果具有很好的精确性。

表 4　结果对比

参数	Intel 手册	CPU-Z	采集结果
*L*1 cache 大小	16 KB	16 KB	16 KB
*L*1 cache 行长	N/A	64 B	64 B
*L*1 cache 组相联	8	8	8
TLB 大小	N/A	N/A	64

6　结束语

并行计算模型的发展需要辅助程序对模型参数进行动态的获取和分析。本文介绍的并行计算模型参数动态采集分析软件包能够对硬件平台固有的静态参数信息和程序运行的动态信息进行综合检测，从而减轻并行计算模型分析的工作量，提高模型本身的精确性和复杂性。基于 PAPI 的存储层次参数采集方法，在可移植性和易用性上有很好的优势，是一种快速精确的获取存储层次参数的有效途径。

今后要逐步完成软件包其他模块特别是动态数据采集模块的设计工作，另外开发高效的数据自动分析工具也是下一步工作的重要内容。

参考文献

[1] Zhang Yunquan, Chen Guoliang, Sun Guangzhong, et al. Models of Parallel Computation: A Survey and Classification[J]. Frontiers of Computer Science in China, 2007, 1(2): 156-165.

[2] Chilimbi T M, Hill M D, Larus J R. Cache-conscious Structure Layout[C]//Proc. of the ACM SIGPLAN Conference on Programming Language Design and Implementation. New York, USA: ACM Press, 1999: 1-12.

[3] KOJAK Group. KOJAK Patterns[EB/OL]. (2007-10-31). http://www.fz-juelich.de/jsc/kojak/performance_props.

[4] Asanovic K, Bodik R, Catanzaro B C, et al. The Landscape of Parallel Computing Research: A View from Berkeley[R]. EECS Department, University of California, Berkeley, USA, Tech. Rep.: UCB/EECS-2006-183, 2006-12-18.

[5] Saavedra R H, Smith A J. Measuring Cache and TLB Performance and Their Effect on Benchmark Runtimes[J]. IEEE Transactions on Computers, 1995, 44(10): 1223-1235.

[6] Yotov K, Pingali K, Stodghill P. Automatic Measurement of Memory Hierarchy Parameters[C]//Proc. of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. New York, USA: ACM Press, 2005: 181-192.

[7] Dongarra J, Moore S, Mucci P, et al. Accurate Cache and TLB Characterization Using Hardware Counters[C]//Proc. of the International Conference on Computational Science. Heidelberg, Germany: Springer, 2004: 432-439.

[8] 侯晓吻, 张云泉. 基于 PAPI 并行程序性能数据收集、显示与分析的工具软件包框架的研究[C]//2005 中国计算机大会论文集. 北京: 清华大学出版社, 2005.

[9] Intel Corporation. Intel Pentium 4 Processors 570/571, 560/561, 550/551, 540/541, 530/531 and 520/521 Supporting Hyper-threading Technology[EB/OL]. (2005-05-10). http://www.intel.com/design/Pentium4/datashts/302351.htm.

编辑　顾逸斐