

# 网络计算机典型应用程序的 $d$ -TLB 行为分析\*

曲宁 袁鹏 管雪涛 程旭

(北京大学微处理器研究开发中心,北京大学计算机科学技术系,北京,100871; <sup>1)</sup> 通讯作者, E-mail: gxt@mprc.pku.edu.cn)

**摘要** 网络计算机是瘦客户计算模式下的一种交互式信息设备,其上的典型应用程序行为分析对于处理器设计、系统开发有着重要意义。本文基于北大众志网络计算机平台,详细分析了  $d$ -TLB 结构和页面大小等配置参数的变化,对典型应用程序  $d$ -TLB 失效率 and  $d$ -TLB 失效导致的性能损失等所产生的影响。实验结果阐明了北大众志系统芯片中 TLB 部件在自身结构和多种页面支持等方面的设计原理和优势。

**关键词** 网络计算机;  $d$ -TLB; 应用程序行为分析; 全系统模拟器; 踪迹驱动模拟  
中图分类号 TP 302.7

## Characterizing the $d$ -TLB Behavior of Typical Applications on Network Computer

QU Ning YUAN Peng GUAN Xuetao CHENG Xu

(Microprocessor Research and Development Center, Peking University, Department of Computer Science and Technology, Peking University, Beijing, 100871; <sup>1)</sup> Corresponding Author, E-mail: gxt@mprc.pku.edu.cn)

**Abstract** Network computer is an interactive device in thin client computing environment, and studying the behavior of typical applications on this platform is important to the microprocessor design and system development. Based on PKUnity network computer platform, this paper analyzes the  $d$ -TLB miss rate and performance penalty of many typical applications under different  $d$ -TLB structures and page sizes. The experiment results explain the advantage of TLB design in PKUnity SoC which satisfies the requirement of lower power and low complexity.

**Key words** network computer;  $d$ -TLB; application behavior analysis; full system simulator; trace driven simulation

## 0 引言

TLB 是虚拟存储中的关键部件,每次取指和数据访问都依赖于 TLB 高效的完成虚拟地址到物理地址的转换。随着程序工作集的增加以及处理器和内存之间速度差距的不断扩大,一些研究指出,TLB 失效导致的性能损失相对于 20 世纪 80 年代的 4% 至 5%<sup>[1]</sup>,90 年代的 5% 至 10%<sup>[2]</sup>,现在对很多实际程序已经达到 30% 至 60%<sup>[3]</sup>。网络计算机是瘦客户计算模式下的一种交互式信息设备,它利用本地资源与网络资源相结合,灵活支持基于 Linux 和 Windows 的 C/S 客户端模式、多媒体终端以及 B/S 客户端模式等多种应用模式。本文对网络计算机中典

型应用程序的  $d$ -TLB(数据 TLB)行为进行详细的评测与分析,基于 PKUnity(北大众志)第一、二代系统芯片中 TLB 的设计,探讨如何改善  $d$ -TLB 失效对应用程序性能的影响。实验选取的评测程序包括标准测试集 SPEC95、SPEC2000、MiBench 和 Mplayer,依赖图形与网络功能的应用程序 Konqueror/Embedded、Rdesktop/X 以及自行开发的 Rdesktop/FB。相关研究<sup>[4]</sup>结果表明,即使对于 SPECint 2000 程序,评测过程中如果忽略 OS 的行为也会严重影响评测结果,因此本文实验均采用面向网络计算机平台的 Unichos 全系统模拟器。

本文分析过程参考了 PKUnity 中 TLB 的设计参

\* 国家“863”高技术研究发展计划(2005AA111010)资助项目

收稿日期:2006-03-01; 修回日期:2006-06-16

数,但是分析结果并不依赖于特定体系结构,因此对网络计算机处理器设计有着广泛意义。针对网络计算典型应用程序的  $d$ -TLB 分析结果表明:(1)在北大志系统现有的设计情况下, $d$ -TLB 失效导致的性能损失对部分应用程序运行产生明显的影响。相对于通用处理器来说,实验中选取的评测程序(如 SPECint 95 reference)数据集规模很小,但  $d$ -TLB 失效导致的性能损失经常达到 4%~7%,甚至会超过 20%。随着北大志系统主频的提高和 cache 等部件的优化,可以预见  $d$ -TLB 失效对应用程序性能的影响还将进一步提高。(2)日常应用类程序通常  $d$ -TLB 失效率很低,并且几乎不受数据集大小的影响,这与程序中的编解码算法常针对固定大小数据块操作的特点有关。这类程序中  $d$ -TLB 失效对性能的影响可以忽略。(3)扩大页面大小可以普遍、有效的降低  $d$ -TLB 失效率。但由于网络计算机中应用程序工作集通常很小,因此支持若干种小型页面(如 16,64 K),已经可以明显改善  $d$ -TLB 对性能的影响。但是同时管理多种页面需要操作系统提供有效的支持。(4)多进程特性并不一定是导致  $d$ -TLB 失效率较高的原因(例如 Konqueror)。因此在 TLB 中支持进程号信息并不会改善某些多进程应用程序的性能。

本文第 1 部分介绍应用程序  $d$ -TLB 行为分析的相关工作,并简要说明了全系统模拟器在研究工作中的重要性。第 2 部分介绍了网络计算机典型应用程序的选择、硬件平台及实验中使用的模拟环境。第 3 部分详细分析了典型应用程序  $d$ -TLB 行为的评测结果。文章最后总结了网络计算机中典型应用程序  $d$ -TLB 行为的特点,并对将来的工作提出了建议。

## 1 相关工作

由于 TLB 自身的重要性以及对系统性能的影响,有大量研究工作<sup>[1]</sup>针对 TLB 硬件设计进行改进。90 年代末期以来,研究人员开始认识到分析各种复杂应用程序的行为对于改善 TLB 设计至关重要。Kandiraju<sup>[5]</sup>等人从多个方面对 SPEC2000 浮点和整点测试集的  $d$ -TLB 行为进行细致分析,为如何改善  $d$ -TLB 失效提供了大量的参考数据和建议。此研究针对高端通用处理器,因此更加关注软件 TLB 失效处理、TLB 预取等特性。而本文则重点考虑网络计算机中、低端处理器的需求,改善  $d$ -TLB 失效对程序性能造成的影响,同时通过保持简洁的 TLB 设计以满足低复杂度和低功耗等需求。Sohoni 等人<sup>[6]</sup>就近年来广泛应用的多媒体程序的访存行为进行了

详细分析,提出多媒体程序的编解码特性使其比 SPEC 程序能够更好的利用 cache 和 TLB。还有一些研究针对 Java 等有特色的程序进行分析<sup>[7,8]</sup>。本文基于网络计算机平台,评测程序不仅包括 SPEC 等标准评测程序,还选择了有代表性的实际应用程序,使得分析结果能够反映 TLB 设计上的变化对实际应用程序性能产生的影响。

应用级模拟器的局限性限制了评测程序的范围,也使得评测结果不够真实<sup>[4]</sup>,近年来全系统模拟器在研究领域得到了广泛重视,例如早期的全系统模拟器 SimOS<sup>[9]</sup>,Simics,以及近两年出现的网络全系统模拟器 M5<sup>[10]</sup>和面向 ARM 处理器的全系统模拟器 ss-os(包括在 SimpleScalar<sup>[11]</sup> 4.0 工具集中)。本文使用的全系统模拟器 Unichos 正是为了更加深入研究和分析网络计算机系统而开发的。实验基于 Unichos 模拟器,采用执行驱动和踪迹驱动互为补充的模拟方法,能够支持对 Konqueror 等复杂应用程序行为进行细致的分析。

## 2 实验环境

### 2.1 应用程序集选择

网络计算机作为新兴的领域还没有公认的标准测试集,因此本文针对北大志网络计算机的应用模式,选定 3 类应用程序作为北大志网络计算机的测试集。由于浮点协处理器在很多中低端处理器中是可选的,本文仅选择整点评测程序。所有程序缺省编译选项为 `-O2 -msoft-float -static`,其中 `msoft-float` 表示程序中所有浮点操作采用软件模拟浮点库实现。

第 1 类评测程序针对处理器和存储系统,本文选择了代码和数据规模较小的 SPECint 95 (reference 输入集)和 SPECint 2000 (train 输入集)。在主频为 150 MHz 的北大志网络计算机上,上述 SPEC 程序的运行时间从几秒到几十分钟。新一代北大志网络计算机的主频已经超过 300 MHz,因此针对此程序集的研究结果对未来的处理器设计也有指导意义。

第 2 类是日常类应用程序代表,参考 Mibench 第一版测试集,去除所有采用浮点算法的程序以及动态指令数少于  $1 \times 10^8$  的程序,并加入 Mplayer 作为多媒体视频测试程序,文中对这个测试集简称为 MibenchINT+,测试程序分 4 类:消费电子类(jpeg, mad, Mplayer),网络算法类(dijkstra, patricia),安全类(blowfish, rijndael)和通讯类(adpcm, crc)。

第 3 类选择网络计算机上行为相对比较复杂的

关键应用程序,包括基于 Qt/Embedded 库的浏览器 Konqueror,基于 X server 的 Rdesktop/X,和基于 Frame Buffer 设备的 Rdesktop/FB。Konqueror 的测试中采用 iBench 性能测试网页集,通过 java script 脚本驱动自动浏览 60 张网页。Rdesktop 测试用例包括浏览 iBench 网页,浏览 word 文档以及浏览图文混合的 ppt 文档,整个测试过程均使用脚本驱动,自动完成登录 Windows 服务器、打开程序和翻页等操作。

以上 3 类评测程序涵盖了网络计算机中现有的大部分整点应用类型,从简单的微型评测程序到相对复杂的实际应用程序。对这些评测程序的分析结果有助于更好理解网络计算机上应用程序的行为。

### 2.2 实验平台和环境

PKUnity I 和 PKUnity II 是面向网络计算机设计的两款高性能、高集成度、低功耗的系统芯片。内部集成了自主指令系统的 32/16 位定点微处理器 Unicore 32、64 位浮点协处理器 Unicore-F64,支持通用操作系统的存储管理单元(MMU),完整的 cache 和 TLB 功能。此外还集成了 PCI 2.2 总线桥控制器、10/100 Mbps 自适应以太网卡控制器、外部存储控制器等。芯片中 d-TLB 相关参数如下:

PKUnity :分离的一级 TLB 设计;d-TLB 为 64 项,4 路组相联。

PKUnity :分离的两级 TLB 设计;d-TLB 第 1 级为 8 项,全相联;第 2 级为 64 项,4 路组相联。

注:d-TLB 表项替换均采用 round-robin 算法;失效时直接访问主存,读取一个页表项需要 20 个时钟周期。

目前 PKUnity 系统芯片中尚不支持详尽的性能计数器,因此本文实验主要基于 Unichos 全系统模拟器开展。为了高效的分析 d-TLB 配置参数变化对应用程序 d-TLB 失效率的影响,本文实验还采用踪迹驱动模拟的方法。参照 SimpleScalar 中的 simr-cache 实现了 simr-tlb 模拟器,可以针对全系统模拟器生成的多进程踪迹信息进行分析。Unichos 模拟器的硬件功能和参数配置与 PKUnity I 网络计算机基准平台完全一致,模拟器中运行的软件环境也保持一致,操作系统为 PKUnity Linux 2.4.17,应用程序均在 NFS 环境中运行。

## 3 实验结果

### 3.1 d-TLB 失效初步分析

首先基于 Unichos 模拟器,通过执行驱动模拟来分析所有评测程序的 d-TLB 失效率以及 d-TLB 失效导致的性能损失。本节中 Unichos 模拟器采用 PKUnity I 中的 TLB 配置参数(详细参数见 2.2 节)。实验结果如表 1 所示。

表 1 应用程序 d-TLB 失效率和 d-TLB 失效造成的性能损失

Table 1 Applications' d-TLB miss rate and performance penalty by d-TLB miss

单位: %

| SPECint 95 (reference 输入集) |      |       | SPECint 2000 (train 输入集) |      |       | MibenchINT+ |      |      |
|----------------------------|------|-------|--------------------------|------|-------|-------------|------|------|
| 程序                         | 失效率  | 性能损失  | 程序                       | 失效率  | 性能损失  | 程序          | 失效率  | 性能损失 |
| go (null)                  | 1.14 | 7.79  | mcf                      | 9.28 | 21.04 | patricia    | 0.65 | 2.52 |
| go (9to21)                 | 1.14 | 7.78  | crafty                   | 2.37 | 9.31  | adpcm       | 1.33 | 4.36 |
| go (5to21)                 | 1.07 | 7.36  | parser                   | 0.74 | 5.38  | mpegpes     | 0.74 | 4.64 |
| gcc (insn)                 | 0.75 | 4.53  | perl (perfect)           | 0.76 | 5.01  |             |      |      |
| compress                   | 1.43 | 11.01 | gap                      | 0.96 | 6.78  |             |      |      |
| perl                       | 0.49 | 4.80  | vortex                   | 1.05 | 8.72  |             |      |      |
| vortex                     | 1.42 | 10.38 | bzip2                    | 0.74 | 7.31  |             |      |      |
| Rdesktop/X                 |      |       | Rdesktop/FB              |      |       | Konqueror   |      |      |
| 程序                         | 失效率  | 性能损失  | 程序                       | 失效率  | 性能损失  | 程序          | 失效率  | 性能损失 |
| xweb                       | 0.85 | 4.28  | web                      | 0.50 | 3.13  | k1          | 0.71 | 6.70 |
| xword                      | 0.96 | 4.54  | word                     | 0.55 | 3.81  | k2          | 0.75 | 7.10 |
| xppt                       | 0.89 | 6.10  | ppt                      | 0.41 | 3.47  |             |      |      |

前两栏中列出的 SPECint 95 和 SPECint 2000 程序,均具有 d-TLB 失效率超过 0.5%,并且 d-TLB 失效导致的性能损失超过 4% 这两个特征。其中 SPECint 95 中,gcc (insn) 的 d-TLB 失效率在 gcc 相关的 15 个测试程序中最高。实验表明,在 SPECint 95 reference 的 24 个测试程序中,有 10 个满足条件(gcc

占 4 个)。在 SPECint 2000 train 的 16 个测试程序中,有 8 个满足条件(perl 占 2 个)。总体来看,接近一半程序中 d-TLB 失效导致的性能损失超过 4%,最高的 mcf 达到了 21.04%。表 1 中第 1 类评测程序 d-TLB 导致的性能损失平均为 7.54%,说明 d-TLB 对此类应用程序的性能造成了明显的影响。



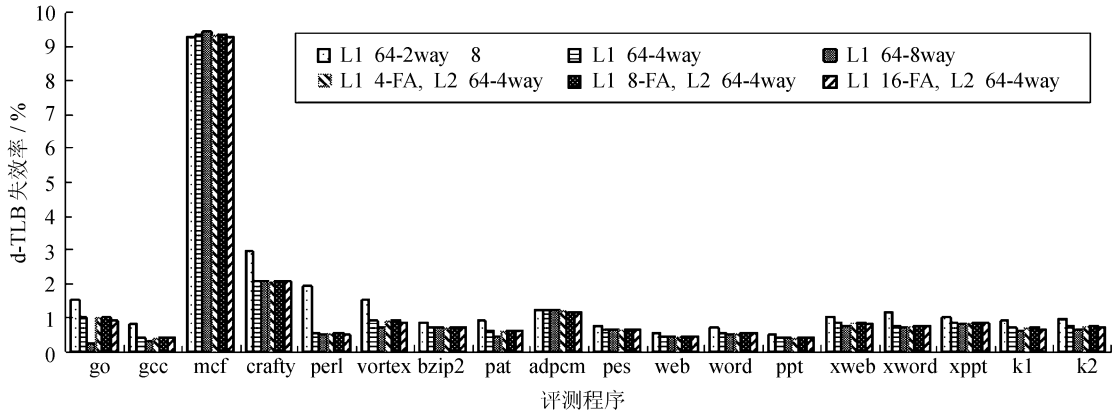


图 1 采用不同的 TLB 结构配置参数时,典型应用程序 d-TLB 失效率统计

Fig. 1 Applications' d-TLB miss rate under different TLB structures

量);另一方面是两级 TLB 之间数据有所冗余。因此 PKUnity 中添加的第 1 级 TLB 并不能有效的降低 d-TLB 失效率,不过因为第 1 级 TLB 表项很少,却更容易实现全相联的设计,同时 TLB 命中的时延也会有所改善。分析结果还表明,在 2 级 TLB 设计中,当第 1 级 TLB 从 4 项增加到 8 项、16 项时,可有效改善自身的失效率。第 1 级 TLB 容量加倍时,其自身的失效率平均改善了 40% 以上。结果说明在可能的情况下,扩大第 1 级 TLB 的大小仍然非常重要。

### 3.3 不同页面大小的影响

使用更大的页面几乎不会增加硬件的复杂度,而且能够有效降低 TLB 失效率。高端服务器需要支持大规模的科学计算,因此倾向于采用更大的页面,如 IA-64 中最大的页面为 256 MB。但是增加页面大小会导致程序内存需求随之增加,研究<sup>[12]</sup>表明页面增大一倍,应用程序占用的内存可能增加 30%,这常常是不可接受的。因此一般系统中会同时支持

多种页面大小,通过在页表项中包含页面大小的信息,使得一个 TLB 表项可以映射更大的地址范围。对于网络计算机来说,应用程序数据集规模相对较小,通过支持小型页面(如 16 KB, 64 KB 等)来有效的改善 d-TLB 失效率是本节分析的重点。由于踪迹驱动模拟中无法包含操作系统的动态管理信息,因此无法支持多种页面大小的评测。实验采用扩大系统中基本页面大小(16 KB, 64 KB, 256 KB)的方式,其中针对 PKUnity 的两级 d-TLB 设计,仅选择全相联的第 1 级 TLB 支持更大的页面,第 2 级 TLB 仍然仅支持 4 KB 页面。

实验结果如图 2 所示。从图中可以看出,针对 PKUnity I 的 TLB 配置,更大的页面使得 d-TLB 失效率显著降低。采用 4、16、64 和 256 KB 页面时,平均失效率分别为 1.42%、0.59%、0.25% 和 0.06%。实际上,使用 16 KB 页面已经可以有效的降低 d-TLB 失效率,除了 mcf (5.23%) 程序之外,其他所有程序的 d-TLB 失效率都降到 0.50% 以下。使用 64 KB 页面

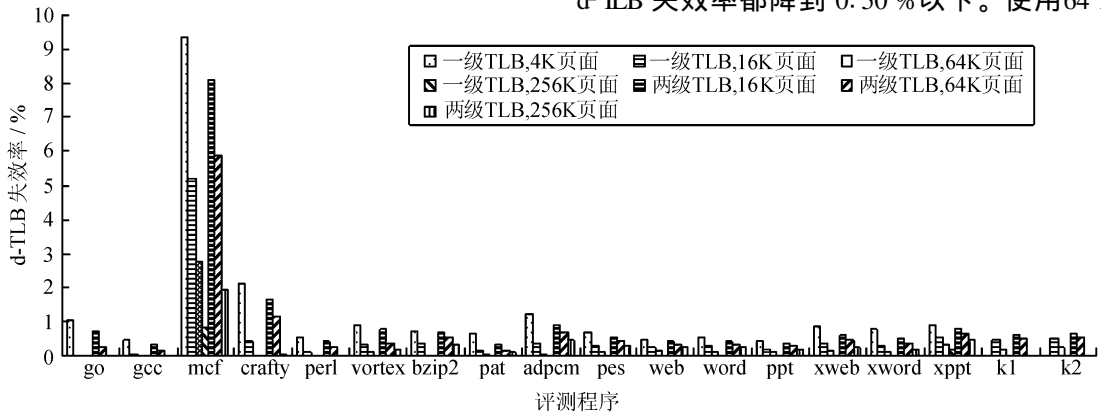


图 2 不同页面配置下应用程序 d-TLB 失效率统计

Fig. 2 Applications' d-TLB miss rate with different page sizes

时,除了 mcf 和 xppt,其他所有程序的 d-TLB 失效率均低于 0.25%,绝大部分程序中 d-TLB 失效导致的性能损失均低于 1%,部分程序甚至接近 0.1%。当使用 256KB 页面时,除了 mcf 和 xppt,所有程序 d-TLB 失效导致的性能损失均低于 0.2%。

针对 PKUnity 的 TLB 设计,使用 16、64 和 256 KB 页面时,平均 d-TLB 失效率分别为 1.07%、0.76% 和 0.33%。由于这个设计方案仅有 8 个一级 TLB 表项可以支持更大的页面,因此失效率的改善情况明显没有前一种方案好,但是从 TLB 失效导致的性能损失角度来看,页面从 16、64 到 256 KB,d-TLB 失效导致的性能损失平均为 5.08%、3.62% 和 1.79%。三类评测程序分别分析,对比 4 KB 和 64 KB 2 种页面配置的情况,第 1、2、3 类应用中 d-TLB 失效导致的性能损失分别改善了 46.63%、50.12% 和 35.26%。同时这一结果也说明具有复杂行为的第 3 类应用程序更难通过简单的硬件方案进行优化。

考虑到内存需求的增加,单纯增加基本页面大小并不实际,但是同时支持多种页面需要 TLB 采用功耗较大的全相联设计。相对于 PKUnity 中的组相联 d-TLB 设计,PKUnity 中 d-TLB 增加了第一级全相联 TLB,可以同时支持多种页面大小,而且 8 个表项也限制了整体功耗的增加。不过个别程序(如 mcf)即使采用 256 KB 页面时 d-TLB 失效对性能的影响仍然超过 4%。因此针对一些大范围数据访问的需求,也需要支持更大的页面。经过权衡,最终 PKUnity 的 d-TLB 设计支持 4 种页面:4、16、64 和 4 MB。这个方案的优势在于:(1) 有效利用 3 种小型页面可以显著改善 d-TLB 失效率,同时应用程序对内存的需求不会有大幅增加;(2) 4 M 页面为支持一些特殊应用需求提供了余地,例如显卡 frame buffer 空间映射等;(3) 整体设计保持其简单的特点,硬

件复杂度和功耗都不会有明显增加。

### 3.4 多进程的影响

使用全系统模拟器进行评测时,执行环境具有多进程的特性,而且第 3 类中的部分应用程序(如 Konqueror)本身就是多进程程序,因此有必要考虑运行过程中多进程特性对 d-TLB 失效率的影响。本节针对 Rdesktop/X(xword, xppt)以及 Konqueror 的行为进行了详细分析。Rdesktop/X 运行过程需要 2 个进程,主进程是 Rdesktop 本身,辅助进程 X server 用来完成绘图等功能。而 Konqueror 则通过一个主进程完成页面分析,显示等功能,同时针对每个页面会启动 1~3 个辅助进程完成网页中相关文件的下载。在全系统模拟器生成的踪迹中,通过跟踪进程号信息能够监视应用程序进程和所有系统进程(线程)的行为,并加以区分。分析结果表明所有程序运行过程中只有 4 个系统线程产生影响,分别是 kupdated、ksofirq-CPU0、keventd 和 rpciod,访存次数总和大部分情况下低于 0.4%,最高也没有超过 1%。而且 Linux 内核中切换到系统线程时会复用之前进程的地址空间,并不会导致 TLB 刷新。因此系统线程的影响在本实验中可以忽略。

具体实验数据如表 2 所示。左半部分是整个测试程序总的访问次数,以及主进程和辅助进程的访存次数、所占比例。xword 测试中主进程并没有占据大部分访存次数,而 xppt 中情况正好相反。这是由于 word 文档以文字信息为主,而 ppt 播放过程中大量背景图片的变化会导致 RDP 协议的数据量大幅增加。Konqueror 在整个测试过程中共启动了 90 个辅助进程,但是所有辅助进程访存次数总和只占 1.54%,平均每个任务执行周期内访存约为  $3 \times 10^6$  次。可见 Konqueror 的辅助进程仅执行一些非常简单的任务,对整个程序几乎没有影响。

表 2 多进程环境下应用程序 d-TLB 失效率统计

Table 2 Applications' d-TLB miss rate under multi-process environment

| 评测程序  | 访问次数           |                |        |             |        | TLB 失效率   |      |      |      |      |
|-------|----------------|----------------|--------|-------------|--------|-----------|------|------|------|------|
|       | 总计<br>访问次数     | 主进程            |        | 辅助进程        |        | 总计<br>TLB | 主进程  |      | 辅助进程 |      |
|       |                | 访问次数           | 所占比例/% | 访问次数        | 所占比例/% |           | mp   | sp   | mp   | sp   |
| xword | 257 851 109    | 46 453 892     | 18.02  | 211 357 072 | 81.97  | 0.76      | 0.47 | 0.46 | 0.82 | 0.81 |
| xppt  | 723 153 865    | 562 347 917    | 77.76  | 159 710 526 | 22.09  | 0.88      | 0.48 | 0.42 | 2.41 | 2.06 |
| k1    | 17 725 966 454 | 17 451 192 692 | 98.45  | 273 305 349 | 1.54   | 0.70      | 0.70 | 0.70 | 0.88 | 0.77 |

表 2 中右半部分是 d-TLB 失效率,分析过程仅对程序访存踪迹的中间部分进行模拟,被模拟的访存踪迹占程序所有访存的 14%~40%。表中的总

计表示正常执行这段踪迹时的 d-TLB 失效率,mp 表示在多进程环境下某个进程自身的 d-TLB 失效率,sp 则假设在运行过程中只有某个进程在运行,忽略

其他所有进程的数据访问及进程切换对其造成的影响。可看到,在 sp 的假设下只有 xppt 的主进程和辅助进程 d-TLB 失效率有明显改善,相对 mp 分别降低了 12.5% 和 14.5%。xword 程序中两个进程的改善都没有超过 3%。Konqueror 评测程序中的辅助进程表示踪迹中某个辅助进程的模拟结果。在 Konqueror 的评测结果中,对辅助进程本身来说,d-TLB 失效率确实有很大改善,但是辅助进程在程序整个执行过程中占有比例非常小;而主进程的 d-TLB 失效率并没有明显改善。这也说明,Konqueror 程序本身虽然是多进程的,但是其较高的 d-TLB 失效率并不是由于多进程这一特性导致的。对于 Rdesktop/X 来说,多进程的影响相对较大。因此在网络计算机中,特别是针对某些特定应用环境(如浏览器/服务器模式下),并不需要在 TLB 加入进程号信息,虽然这一设计可避免多进程之间的干扰,但也会导致 TLB 大小和复杂度有显著的增加。

## 4 结 论

本文对网络计算机典型应用程序的 d-TLB 行为进行深入分析,基于全系统模拟器,采用执行驱动模拟和踪迹驱动模拟相结合的分析方法。实验结果表明,即使在低端的网络计算机中,d-TLB 失效导致的性能损失仍然是系统中不可忽略的因素。而随着北大众志系统芯片主频的提高,Cache 等功能的优化,d-TLB 失效导致的性能损失在应用程序中所占的比例还会进一步提高。因此如何降低 TLB 失效导致的性能损失是网络计算机设计中需要考虑的一个重要因素。本文针对这一问题,首先对不同的 TLB 结构配置参数产生的影响进行了详细分析。此外,结合高端服务器中广泛支持的多种页面技术,通过实验结果说明支持若干种小型页面能够有效降低网络计算机中应用程序的 d-TLB 失效率。针对网络计算机处理器在复杂度、功耗等方面的限制,文中阐明了 PKUnity II 中 TLB 设计的原理和优势。其简洁有效的两级 TLB 设计,从硬件上提供了对多种页面的基本支持,把如何有效管理多种页面,同时避免增加页面大小带来的副作用(例如程序启动和写时拷贝等过程的时延增加)这些问题留给操作系统来处理,这也为网络计算机操作系统的进一步发展提供了探索的空间。本文中所构建的网络计算机全系统模拟评测平台与工具,也为面向网络计算机平台的评测和研究工作提供了坚实的基础。

## 参 考 文 献

- [1] Clark Douglas W, Emer Joel S. Performance of the VAX 11/780 translation buffer: Simulation and measurement. *ACM Transactions Computer Systems*, 1985, 3(1): 31-62.
- [2] Uhlig Richard, Nagle David, Mudge Trevor N, et al. Trap driven simulation with Tapeworm II//Proc of the Sixth International Conference on Architectural Support for Programming Languages and Operating Systems (SIGARCH). New York: ACM Press, 1994: 132-144.
- [3] Navarro Juan. Transparent operating system support for superpages [D]. Houston: Rice University, 2004.
- [4] Cain H W, Lepak KM, Schwartz B A, et al. Precise and accurate processor simulation//Proc of the Fifth Workshop on Computer Architecture Evaluation Using Commercial Workloads, HPCA. Boston: IEEE Press, 2002: 13-22.
- [5] Kandiraju Gokul B, Sivasubramaniam Anand. Characterizing the d-TLB behavior of SPEC CPU2000 benchmarks//Proc of ACM SIGMETRICS. Marina Del Rey: ACM Press, 2002: 129-139.
- [6] Sohoni Sohum, Xu Zhiyong, Min Rui, et al. A study of memory system performance of multimedia applications//Proc of ACM SIGMETRICS. Cambridge: ACM Press, 2001: 206-215.
- [7] Shuf Yefim, Serrano Mauricio J, Gupta Manish, et al. Characterizing the memory behavior of Java workloads: A structured view and opportunities for optimizations//Proc of ACM SIGMETRICS. Marina Del Rey: ACM Press, 2002: 194-205.
- [8] Li Tao, Kurian John Lizy, Vijaykrishnan Narayanan, et al. Using complete system simulation to characterize SPECjvm98 benchmarks//Proc of the 14<sup>th</sup> International Conference on Supercomputing. Santa Fe: ACM Press, 2000: 22-33.
- [9] Rosenblum M, Herod S, Witchel E, et al. Complete computer system simulation: The SimOS approach. *IEEE Parallel & Distributed Technology*, 1995, 3(4): 34-43.
- [10] Binkert Nathan L, Hallnor Erik G, Reinhardt Steven K. Network-oriented full-system simulation using M5//The Sixth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW). Anaheim: IEEE Press, 2003: 36-43.
- [11] Austin Todd M. SimpleScalar version 4.0 release//Proc of the 34<sup>th</sup> Annual International Symposium on Microarchitecture (micro-34). Istanbul: IEEE Press, 2001: 22-30.
- [12] Talluri Madhusudhan, Kong Shing, Hill Mark D, et al. Tradeoffs in supporting two page sizes//Proc of 19<sup>th</sup> Annual International Symposium on Computer Architecture (ISCA). Queensland: ACM Press, 1992: 415-424.