

龙芯 2 号处理器系统优化关键技术

伍 鸣 张福新 林 伟 许先超 袁 楠 王 剑

(中国科学院计算技术研究所计算机系统结构重点实验室 北京 100080)

(中国科学院研究生院 北京 100049)

(wuming@ict.ac.cn)

Critical Techniques of System Optimization for Godson-2 Processor

Wu Ming, Zhang Fuxin, Lin Wei, Xu Xianchao, Yuan Nan, and Wang Jian

(Key Laboratory of Computer System and Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

(Graduate University of Chinese Academy of Sciences, Beijing 100049)

Abstract As the interface between applications and processor, system software plays an important role in maintaining the stabilization of processor and improving the performance of applications. In this paper, ways to resolve cache synonyms in Godson-2 processor and methods to decrease performance loss resulting from TLB miss, such as augmenting the page size, software TLB and FAST_TLB_REFILL, are described. And a way of Uncache Accelerate for improving performance of video output is depicted. Experimental results show that these kinds of methods in system software can benefit the stabilization and performance of the computer system.

Key words Godson-2 processor; cache synonyms; software TLB; Uncache Accelerate

摘 要 系统软件作为处理器和应用程序之间的接口,对于充分利用处理器的特性来维护处理器与应用程序的稳定性和提高应用程序的性能起着极其重要的作用。描述了在 Linux 内核中解决龙芯 2 号处理器的 Cache 别名问题的方法以及通过增加页的大小、软 TLB 和 FAST_TLB_REFILL 的方法减小 TLB 失效的性能损失,还有 Uncache Accelerate 对媒体播放软件的加速。实验结果表明,在系统软件中增加这些方法的支持,对系统的稳定性和性能都有较大的好处。

关键词 龙芯 2 号处理器;Cache 别名;软 TLB;Uncache Accelerate

中图法分类号 TP316.81

1 引 言

操作系统与处理器体系结构紧密相关,系统软件对处理器的支持主要要解决两个问题:①使应用程序能在处理器上稳定运行;②使应用程序能够充分利用处理器的特性,从而高效地运行。龙芯 2 号

高性能通用处理器(简称龙芯 2 号)^[1]所实现的 Cache 访问方式是用虚地址索引并用物理地址匹配 tag。这意味着龙芯 2 号将遇到经典的 Cache 别名(synonyms)问题^[2,3]。该处理器为了减小硬件代价,并没有在硬件上维护这种 Cache 的一致性。因此,如何解决好虚地址 Cache 带来的这一问题,将是系统软件首要解决的。

收稿日期:2005-07-11;修回日期:2005-12-02

基金项目:国家自然科学基金杰出青年基金项目(60325205);国家“八六三”高技术研究发展计划基金项目(2002AA110010);中国科学院知识创新工程重大项目(KGCX2-109)

由于龙芯 2 号是与 MIPS 体系结构^[4]兼容的,其 MMU 是由软件维护的,因此当 TLB 失效发生时,将由系统软件来实现虚实地址映射的重构工作.在现实的应用中,有很多程序访问的数据量很大,而且局部性不强,这些程序在运行中会产生大量的 TLB 失效.由于 TLB 失效的处理是由系统软件来完成,因此会带来较大的系统开销.

对于 I/O 密集型的应用程序而言,除了 CPU 本身影响性能外,系统的 I/O 性能也是一个重要因素.因此对于此类应用来说,提高 I/O 的效率也是提高程序性能一个重要手段.媒体播放软件就是这样的一个典型的应用.在该软件中,视频输出所占开销的比重非常大.在龙芯 2 号中,我们实现了 uncache accelerate(后面简称为 UA)的功能.其基本思想是使对 I/O 设备上存储器中的数据访问能够成块地处理.因此,充分利用好处理器提供的这种 I/O 支持对于提高媒体播放的性能将有很大的好处.

本文描述在龙芯 2 号的研制中如何利用系统软件支持来解决上述问题.

2 Cache 别名问题及 Linux 上的解决

2.1 Cache 别名问题

在采用虚地址索引和物理地址做 tag 的 Cache 中,存在 Cache 别名问题,这主要是因为不同的虚地址在 Cache 中索引不同的 Cache 行,但却可能映射到相同的物理地址,这使得 Cache 中可能存在相同物理地址所对应数据的两个或多个备份,从而会产生 Cache 一致性问题.然而,并不是所有的虚地址 Cache 结构都会产生 Cache 别名的问题,这和 Cache 的大小、索引和匹配机制、操作系统中页的大小都有密切的关系.

目前龙芯 2 号的 Cache 就采用上述的结构,其大小为 64KB,4 路组相联,每一路 Cache 的大小为 16KB,所以需要用虚地址的低 14b(13..0)去进行 Cache 行的索引.然而在 Linux 等通用操作系统中,内存页的大小被默认为 4KB,所以,虚地址和物理地址的低 12b(11..0)是相同的.若两个虚地址的(13..12)位是不同的(假设为 V_a , V_b),它们将会索引到不同的 Cache 行,但却可能映射到同一个物理地址.如果 V_a 将数据写入 Cache 中,但是 V_b 却无法感知这一行为,之后从 V_b 中读出的数据将是过期的错误数据.

2.2 Linux 中 Cache 别名的解决方法

由于龙芯 2 号并不支持维护 Cache 一致性的硬件解决方法,所以,对一致性的维护就必须由系统软件来完成.在 Linux 内核^[5]中,有 3 种 Cache 别名的解决方法:①改变内存页的大小,使内存页的大小变为 16KB;②改变地址映射机制,使虚地址和其所映射物理地址的(13..12)位(称为颜色位)总是相同,为了方便起见,在后面称之为页着色法;③检测 Cache 别名可能出现的情况,在必要时写回 Cache 并置无效,从而动态地维护 Cache 的一致性,后面称之为动态检测法.

第 1 种方法将页的大小变为 16KB,这意味着虚地址和其所映射的物理地址的低 14b 都是相同的,因此当用虚地址的低 14b 去索引 Cache 时,相当于用物理地址的低 14b 去索引,所以不会产生 Cache 别名.使用这种方法的好处是,由于增加了页的大小,使得 TLB 所能覆盖的地址范围增加了,于是提高了 TLB 的命中率.

实现页着色算法的内核,在为用户进程分配物理页时,保证了映射到该物理页的虚地址与所映射的物理地址的颜色位相同.因此,这样就使得虚地址中用来索引 Cache 的低 14b 与对应的物理地址的低 14b 是相同的,于是避免了 Cache 别名的出现.但是这种实现方法由于加强了地址映射的限制条件,因而增加了内核中伙伴分配算法的复杂度,并且还会带来文件系统中页缓存的频繁的删除和创建,从而降低了系统的性能.

实现了动态检测算法的内核需要在为用户进程建立和断开虚实地址映射时检测 Cache 别名可能出现的情况.如果建立映射时,虚实地址的颜色位不同,则在映射建立之前,内核需要通过内核虚地址将 Cache 中的内容写回并置无效.反之,当断开映射时,如果虚实地址的颜色位不同,则在映射断开之前,内核需要通过用户进程的虚地址将 Cache 中的内容写回并置无效.由此可见,在动态检测算法中,Cache 的写回和置无效操作将成为主要的性能开销,并且由于频繁刷新 Cache 会增加 Cache 的失效次数,从而影响系统的性能.

3 减少 TLB 失效所带来的性能损失

减少 TLB 失效所带来的性能影响可以从两个方面去考虑,一个方面是减少 TLB 失效的次数,另一个方面是减少失效本身的开销.

3.1 增加页的大小以减少 TLB 失效次数

增加页的大小可以增大 TLB 所覆盖的内存范围,从而减少 TLB 失效次数.但是操作系统对内存的管理是以页为单位的,这就意味着采用大页的内核中可能出现更多的内存碎片,降低内存的利用率.在操作系统中,当父进程创建子进程时,父进程的内存页可能会被拷贝到子进程相应的内存页中.此时,如果该内存页中的碎片较多将会带来较多的额外开销.另外,在文件系统中,内核对页缓存的管理也是以页为单位的.如果文件数据的大小并不是页大小的倍数,则在页缓存中也将会出现碎片.

我们分别实现了 4KB、16KB 和 64KB 页大小的内核,并用基准测试程序 SPEC2000^[6,7] 和 dbench

在龙芯 2 号上进行了测试.处理器主频为 90×4.5 (405) MHz,内存大小为 512MB, L1 缓存大小为 64KB,相联度为 4 路组相联, L2 缓存大小为 8MB,操作系统为 64b Linux,其版本号为 2.4.22.

(1) 不同页大小对 SPEC2000 基准测试程序的影响

图 1 是在 3 种不同页大小的系统下测得的 SPEC2000 定点程序和浮点程序的分值.由该图可以看出,大页系统对于 SPEC2000 的 50% 左右的程序有较大的性能提高.对于定点程序来说,大页系统的效果更加明显.在 64KB 页的系统下,定点程序的总分比 16KB 页系统提高了 5% 左右,而浮点程序的总分却只能提高不到 1%.

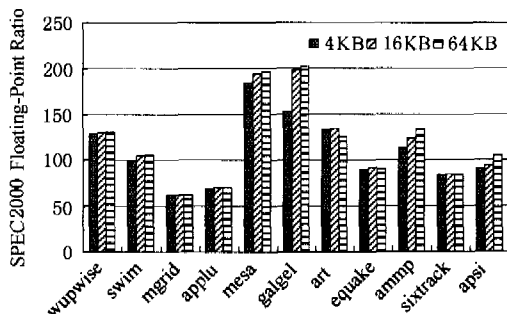
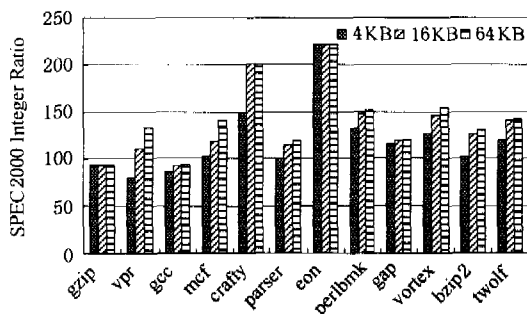


Fig. 1 Integer ratio and floating-point ratio of SPEC2000 in systems with different page size.

图 1 不同页大小系统中的 SEC2000 的定点分值和浮点分值

为了更加详细地分析 SPEC 程序在不同页大小系统中的行为,我们利用龙芯 2 号提供的 Performance Counter^[8] 的功能在 SPEC2000 下测得了一些性能数据.图 2 中列出了 SPEC2000 中比较典型的几个程序在不同页大小系统中的 TLB 重填例外次数的变化.由于大页系统对应用程序所带来的直接好处就是减少了 TLB 重填例外的次数,因此,TLB 重填例外次数较多的程序所受到的影响也就相对要大一些,比较明显的是 vpr, mcf, bzip2,

galgel, ammp 和 apsi.与定点程序相比,大部分的浮点程序 TLB 重填例外的次数较少,因此,大页系统对于浮点程序的影响不大.其中,apsi 是个例外.即使在 64KB 页的系统中,它的 TLB 失效次数也相当多,这说明更大页的系统对它仍然会有较大影响.

另外,定点程序中有两个较为特殊的程序 crafty 和 twolf.这两个程序的分值从 4KB 页系统到 16KB 页系统有明显的提高,但从 16KB 页系统到

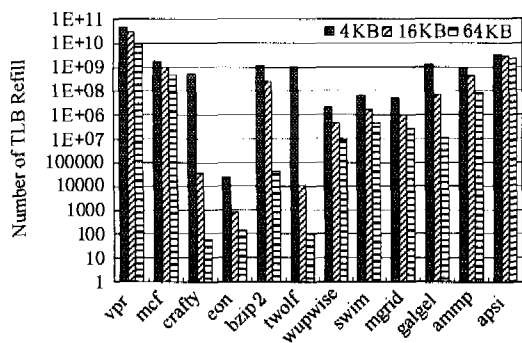


Fig. 2 Number of TLB refill in SPEC2000.

图 2 SPEC2000 的 TLB 重填例外次数

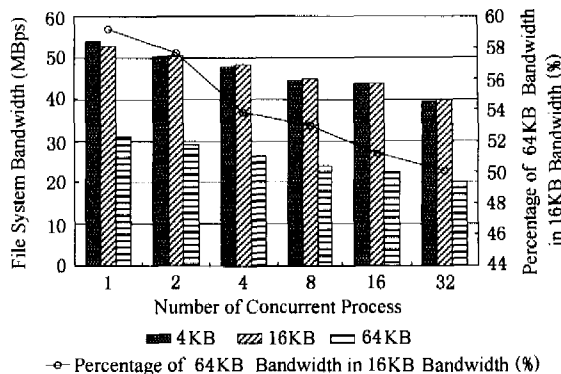


Fig. 3 dbench performance in systems with different page size.

图 3 不同页大小的系统对 dbench 的影响

64KB 页系统却变化非常小. 由图 3 可以看出, 当页大小从 4KB 变为 16KB 时, 这两个程序的 TLB 重填例外的次数骤然下降. 这是因为当系统页大小变为 16KB 时, TLB 的覆盖范围已经接近了这两个程序访存行为的边界, 因此, 这两个程序在 16KB 页大小时, 就只有很少的 TLB 失效了, 于是, 64KB 页的系统也就不会对这两个程序带来太多的好处.

(2) 不同页大小对 dbench 测试程序的影响

由于文件系统中页缓存是以页的大小为基本单位的, 所以磁盘 IO 经常一次传输一个页的数据. 如果对文件数据的访问是跳跃进行的, 并且每次只访问一个页中很少一部分的数据, 同时该数据又没有在页缓存中命中, 则页的大小越大进行的冗余的 IO 传输越多, 因此, 如此行为的文件操作的性能就会降低.

dbench 是开源社团提供的一个通用的、具有代表性的测试文件系统性能的基准测试程序. 它模拟了 Samba 服务器在运行时所产生的文件 IO 操作.

从图 3 可知, 4KB 系统与 16KB 系统的带宽不相上下, 而 64KB 系统的带宽却有了明显的下降. 这说明在实际的应用中, 64KB 页所带来的冗余 IO 的影响已经非常明显. 而且随着并发进程数的增加, 64KB 系统性能降低的比例越大, 这是由于多个进程共享系统中的页缓存, 从而产生竞争的结果.

3.2 通过软 TLB 和 FAST_TLB_REFILL 减少 TLB 失效开销

由于龙芯 2 号是一个 64b 的通用处理器, 兼容 MIPS III 指令集, 因此, 我们在龙芯 2 号上使用 64b 的 Linux 操作系统 (Linux/mips64). 64b 的 MIPS 处理器上的虚地址空间达到了 40b, 所以 Linux/mips64 采用的是三级页表结构, 内核在对 TLB 重载入例外处理时, 必须依次查找三级页表来得到相应的页表项. 在这种页表结构中, 每一次 TLB 重载入 (refill) 例外处理时共需 5 次访存, 可能产生 2~4 次 Cache 失效, 从而使得 TLB 重载入异常的处理时间大大加长.

(1) 通过软 TLB 减少 TLB 失效开销

软 TLB^[9,10] 是在系统中建立一个全局的硬件 TLB 的缓冲区. 当发生 TLB 重载入例外时, 例外处理程序首先在软 TLB 中查找, 如果软 TLB 命中, 则只需要 3 次访存, 而且由于软 TLB 的数据位于连续的地址内, 所以只会产生 1 次 Cache 失效; 如果软

TLB 不命中, 则处理程序需要从三级页表中将物理地址取回并更新软 TLB. 由此可见, 如果软 TLB 不命中则会带来更大的开销, 因此软 TLB 能否达到优化的目的主要在于软 TLB 的命中率, 而影响软 TLB 命中率的主要参数就是软 TLB 的缓冲区的大小和软 TLB 表项中的关联关系.

实验测试表明, 在软 TLB 项数相同的情况下, 软 TLB 的命中率大体上是全相联大于两路组相联, 而两路组相联又大于直接相联. 但是由于在 TLB 重载入例外的处理函数中可用的寄存器只有 k_0 和 k_1 , 所以如果没有硬件上的支持, 实现复杂的相联关系需要增加很多指令来完成, 反而造成了一定的开销. 由于软 TLB 的表项增加到 1KB 以后, 各种相联关系的软 TLB 的命中率的差别不大, 几乎都达到 90% 以上, 所以最终我们采用的是直接映射. 软 TLB 的命中率是随着表项的增多而提高的. 当软 TLB 的表项大于 4KB 后, 其命中率就可以达到 90%. 虽然继续增加软 TLB 的表项数目, 其命中率将进一步上升, 但是意义已经不大, 并且随着表项的增多会造成内核占用更多的内存资源, 增加了内存压力, 使得系统性能下降, 所以我们选择 4KB 的表项数作为软 TLB 的表项.

我们用 4KB 表项直接映射的软 TLB 在龙芯 2 号上运行 SPEC2000 测试程序. 软 TLB 对 SPEC2000 测试程序的分数提高如图 4 所示:

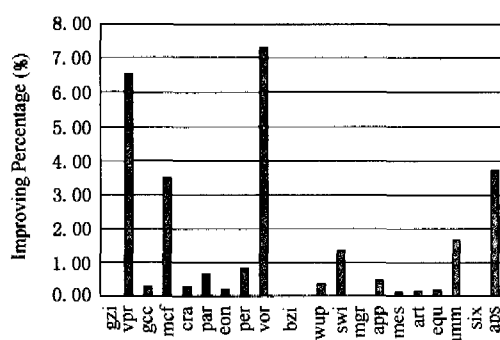


Fig. 4 Improving percentage of software TLB for SPEC2000.

图 4 软 TLB 对 SPEC2000 分数的提高率

(2) 通过 FAST_TLB_REFILL 减少 TLB 失效开销

FAST_TLB_REFILL 方法通过缓存第三级页表指针, 在 TLB 重载入例外处理时可以从这个缓存中直接得到第三级页表的指针, 从而跳过对前两级页表的访问, 其原理如图 5 所示. 其中 *global_hash_ptes* 中存放的是指向当前进程第三级页表的指针.

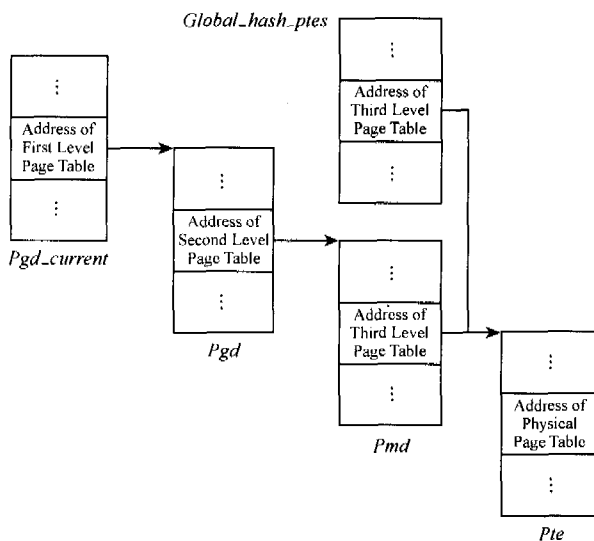


Fig. 5 Principle of FAST_TLB_REFILL.

图5 FAST_TLB_REFILL原理简图

FAST_TLB_REFILL方法中主要的开销在于进程切换时对 *global_hash_ptes* 的维护. 因为 *global_hash_ptes* 是一个全局的结构, 在每次进程切换时, 需要将新进程的第三级页表指针缓存到该结构中. 我们利用 Linux 进程在地址分布上的规律来减小这种开销. 在 Linux 系统中, 程序在地址空间的分布上具有一定的规律性: 代码段一般映射到虚地址的起始部分, 而栈数据放在虚地址的最高部分, 另外库文件也被映射到一个固定地址开始的空间. 因此, 在进程切换时并不用将所有的第三级页表指针都拷贝到 *global_hash_ptes* 中, 而只需要拷贝从一些固定地址开始的几个第三级页表指针, 这样 FAST_TLB_REFILL 的开销就可以减小.

我们选择了 SPEC2000 中 TLB 例外比较频繁的程序进行测试. 实验结果表明, 当程序的 TLB 重载入例外比较频繁时, FAST_TLB_REFILL 可以使

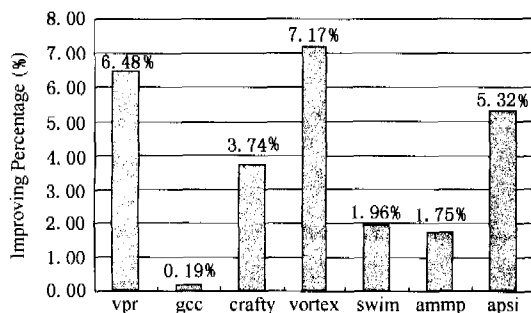


Fig. 6 Improving percentage of FAST_TLB_REFILL for SPEC2000.

图6 FAST_TLB_REFILL对SPEC2000分数的提高率

程序性能得到 1%~7% 的提高, 面向对象的数据库程序 255.vortex 可以提高 7.17%, 如图 6 所示.

4 媒体播放的 UA 支持

媒体播放是典型的 I/O 密集型应用, 其中视频输出所占的开销非常大. 在 MIPS 体系结构中, 由于显存映射在 Uncache 地址空间, 一条 Uncache 访存指令就必然带来一次系统总线事件, 导致系统总线负担重效率低, 并阻塞后续访存指令. 注意到各种应用中连续的 Uncache 访存操作往往是对连续地址访问, 龙芯 2 号的设计者在处理器访存接口处增加了一个 32B 长度的 Uncache 缓冲区, 收集对连续地址的 32b 和 64b 数据的 Uncache 写操作, 缓冲区数据满时以总线上一次突发写 (burst write) 事件将内容写回目标, 以节省带宽, 称为 Uncache Accelerate (UA) 技术. 一次普通的 32b 或 64b 数据的 Uncache 写操作需要 2 个系统周期 (地址数据各占 1 个系统周期), 一次突发写操作需要 5 个系统周期 (1 个周期给出地址, 4 个周期传送数据) 写回 32B 数据, 不考虑北桥与 CPU 访存接口间的流量控制, 使用 UA 后, 对连续地址的 Uncache 写操作的加速比为 1.6~3.2.

对于映射的地址空间, 处理器依据 TLB 项中 Cache Algorithm 域区分以何种 Cache 方式访问物理地址. 对于 Uncache 访问, 我们既可以在 Cache Algorithm 域中指定为 Uncache 模式 (访问数据时就不通过 Uncache 缓冲区), 也可以指定为 UA 模式.

Linux 通过 mmap 系统调用将进程的虚地址映射到显存的物理地址, 以使进程可以通过访问虚存地址的方式来访问显存. TLB 表项来源于操作系统维护的页表项, 进行优化的基本方法是在进程映射显存地址, 建立相应的页表项时, 修改这些页表项的 Cache Algorithm 域为 UA 模式. 经过 UA 对视频输出的优化, MPlayer 媒体播放器的性能得到很大的提高, 视频输出的时间提高了 1~2 倍.

5 结 论

本文从稳定性和性能的角度, 描述了龙芯 2 号处理器系统优化的关键技术. Cache 别名是任何采用虚地址 Cache 的体系结构所必须面临的问题. 在

4KB 页大小的内核上,我们可以采用动态检测法或页着色法。动态检测带来更多的 Cache 刷新操作,增加了 Cache 失效次数,而页着色限制了虚实地址的映射,同时增大了页分配的伙伴算法的复杂度。与 4KB 页的系统相比,16KB 页的系统具有明显的性能优势,而且很自然地解决了 Cache 别名的问题,因此是一种较优的解决方法。然而它也带来了更多的内存碎片。因此,对于不同的应用,采取何种页大小,应该予以不同的权衡。由于龙芯 2 号处理器的 MMU 是由软件来维护的,因此,由 TLB 失效所引起的性能开销比较大。64KB 页系统可以减少 TLB 失效的次数,软 TLB 和 FAST_TLB-REFILL 可以减少 TLB 失效本身的开销,从而提高程序的性能。通过 UA 对媒体播放软件的加速,更加说明了系统支持和处理器结构的密切关系。随着龙芯处理器的不断改进,龙芯的系统支持必将对龙芯处理器的应用起到更大的推动作用。

参 考 文 献

- 1 Hu Weiwu, Zhang Fuxin, *et al.* Godson Processor Project Data. Beijing: Institute of Computing Technology, Chinese Academy of Sciences, 2004 (in Chinese)
(胡伟武, 张福新, 等. 龙芯项目资料. 北京: 中国科学院计算技术研究所, 2004)
- 2 Michel Ceklev, Michel Dubois. Virtual-address caches. Part 1: Problems and solutions in uniprocessors. IEEE Micro, 1997, 17 (5): 64~71
- 3 J. R. Goodman. Coherency for multiprocessor virtual address caches. The 2nd Int'l Conf. Architectural Support for Programming Languages and Operating Systems, Palo Alto, CA, 1987
- 4 Dominic Sweetman. See MIPS Run. San Francisco: Morgan Kaufmann, 1999
- 5 Linus Torvalds, *et al.* Documentation for Linux kernel, version 2.4.22. <http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.22.tar.bz2>, 2003
- 6 The Standard Performance Evaluation Corporation. <http://www.spec.org/cpu2000>, 2000
- 7 Gokul B. Kandiraju, Anand Sivasubramaniam. Characterizing the d-TLB behavior of SPEC CPU2000 benchmarks. USENIX Symposium on Operating Systems Design and Implementation, Boston, MA, 2002
- 8 Joe Heinrich. MIPS R10000 User's Manual. Mountain View, California: MIPS Technologies, Inc., 1996
- 9 Kavita Bala, M. Frans Kaashock, William Weihl. Software prefetching and caching for translation lookaside buffers. USENIX Symposium on Operating Systems Design and Implementation,

Monterey, CA, 1994

- 10 Elphinstone, Heiser. Page tables for 64-bit computer systems: [Ph. D. dissertation]. New South Wales: School of Computer Science & Engineering, University of New South Wales, 1999



Wu Ming, born in 1978. Received his B. A's degree in computer science from the University of Science and Technology of China, Hefei, China, in 2002. Since 2004, he has been a Ph. D. candidate in the Institute of Computing Technology, the Chinese Academy of Sciences, Beijing, China. His current research interests include computer architecture and operating system.

伍鸣, 1978 年生, 博士研究生, 主要研究方向为计算机体系结构、操作系统。



Zhang Fuxin, born in 1976. Received his B. S. degree in Computer Science from the University of Science and Technology of China in 2000 and his master degree in computer science from the Institute of Computing Technology, the Chinese Academy of Sciences in 2002, respectively. He is currently Ph. D. candidate in the Institute of Computing Technology. His main research interests include high performance computer architecture and operating system.

张福新, 1976 年生, 博士研究生, 主要研究方向为高性能计算机体系结构和操作系统 (fxzhang@ict.ac.cn)。



Lin Wei, born in 1980. Received his B. A's degree in computer science from the University of Science and Technology of China, Hefei, China, in 2002. Since 2002, he has been M. A. degree candidate in the Institute of Computing Technology, the Chinese Academy of Sciences, Beijing, China. His current research interests include computer architecture and operating system.

林伟, 1980 年生, 硕士研究生, 主要研究方向为计算机体系结构、操作系统 (wlin@ict.ac.cn)。



Xu Xianchao, born in 1979. Received his B. A's degree in computer science from the University of Science and Technology of China, Hefei, China, in 2002. Since 2002, he has been M. A. candidate in the Institute of Computing Technology, the Chinese Academy of Sciences, Beijing, China. His current research interests include computer architecture and operating system.

许先超, 1979 年生, 硕士研究生, 主要研究方向为计算机体系结构、操作系统 (xuxianchao@ict.ac.cn)。



Yuan Nan, born in 1982. Received his B. A's degree in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2004. Since 2004, he has been M. A. candidate in the Institute of

Computing Technology, the Chinese Academy of Sciences, Beijing, China. His current research interests include computer architecture and operating system.

袁楠, 1982年生, 硕士研究生, 主要研究方向为计算机体系结构、操作系统(yuannan@ict.ac.cn).



Wang Jian, born in 1971. Received his B. S. degree in computer science from the University of Science and Technology of China in 1994 and his M. S. degree in Computer Science from the Institute of

Computing Technology, the Chinese Academy of Sciences in 1997, respectively. He is currently associate professor in the Institute of Computing Technology. His main research interests include high performance computer architecture and operating system.

王剑, 1971年生, 硕士, 副研究员, 主要研究方向为高性能计算机体系结构和操作系统(jw@ict.ac.cn).

Research Background

Operating system has a great effect on the stabilization and the performance of applications and processor. In this paper, concentrating on Linux operating system and Godson-2 processor, several methods that can improve the stabilization and the performance of applications are introduced. They include solutions to cache synonyms, ways to reduce the cost of TLB miss and Uncache Accelerate to improve the performance of video output. Our work is supported by the National Science Foundation of China (60325205), 863 Hi-Tech Research and Development Program of China (2002AA110010) and Knowledge Innovation Engineering Project of the Chinese Academy of Sciences (KGCX2-109).

全国第九届 Java 技术及其应用学术交流会

征文通知

为了总结与交流近年来 Java 的研究、开发与应用的经验,进一步推动我国 Java 技术的发展,为提高我国的信息科学技术的发展水平尽一份力量,由中国计算机学会计算机应用委员会、中国自动化学会计算机应用委员会、中国电子学会计算机工程与应用委员会、中国信息产业商会微型机与应用分会自 1998 年起每年召开一次全国 Java 技术及应用学术交流会,已成功举办了八届,现决定于 2006 年三季度举办全国第八届 Java 技术及其应用学术交流会,主题是“促进企业信息化进程”。我们鼓励企业组织投稿和进行对成熟的产品演示……;欢迎广大信息技术专家、教授、工程师及产业界、企业界、教育界同仁踊跃投稿和参加。

征文重点及范围

①国内外 Java 技术及其应用发展综述及专题报告;②Java 技术产品(包括构件、模块及应用软件)开发及应用成果;③Java 嵌入式技术及信息家电产品开发与应用成果, Jini 技术及其应用成果;④Java 计算、移动计算和分布式计算及其应用经验;⑤Java 技术(J2EE)、开发工具在中间件、各种软件平台开发中的应用成果与经验;⑥Java 技术在电子商务、电子政务、网络通信、企业网络、MIS、ERP、SCM、CRM 中的应用;⑦Java 在各行业的应用成果、经验及典型实例;⑧基于 Java 技术的远程教学、远程诊断与医疗的求解方案;⑨Java 技术在人工智能领域的应用成果;⑩我国 Java 技术发展战略及政策措施建议。

征文要求

论文用 B5 纸打印并附软盘(Word 格式),版芯高 220mm、宽 147mm,按 5 号宋体字排,每页 39 行,每行 40 字;标题 3 号黑体居中排,作者姓名排在标题下,占两行居中排,4 号楷体字,论文摘要用 6 号宋体字排,行宽左右各缩 4 字距,分节标题用 4 号黑体居中排,图和表格可随文排入,但要在版芯尺寸内,一般论文 6000 字,含图不超过 4 页,综述论文可 8000 字,不超过 6 页。

要求:①尚未正式发表的论文;②请另附纸注明作者姓名、电话、传真、Email 地址、通信地址、邮编等,以便联系。

会议将组织专家学者评选优秀论文,给予适当奖励并向核心杂志推荐发表。

论文截止日期:2006 年 7 月 15 日,8 月 30 日将发出录用通知。

时间:2006 年 9 月中旬

地点:北京(具体报到日期、地点另发通知)

来稿请寄:100083 北京 927 信箱 贾志梅 梁钢 龚炳铮 收

或电子邮件:jiazm@ncse.com.cn; gary@teamsun.com.cn; gongbz@ncse.com.cn

联系电话:(010)62327331-115/111 传真:(010)62311179,62325267