

基于多页的龙芯 2F 软 TLB 重载入异常处理改进

何 炜

(江苏自动化研究所,江苏 连云港 222006)

摘 要: 龙芯 2F 系统在 TLB 替换处理上开销较大, 现有软 TLB 技术从减少 TLB 重载入异常处理时间方面考虑, 利用软 TLB 来缓存 TLB 表项, 提高替换处理中 Cached 命中率, 减少了重载入异常处理时间。为进一步提高 TLB 效率, 设计采用多页技术提高 TLB 的覆盖率, 减少 TLB 替换次数, 进而减少重载入异常处理的次数, 达到提高系统性能的目的。

关键词: TLB; 重载入异常处理; 多页技术; 龙芯 2F

中图分类号: TP316.2

文献标识码: A

文章编号: 1674-7720(2011)23-0077-02

Loongson 2F software TLB reload exception handling improved based on multi-page

He Wei

(Jiangsu Automation Research Institute, Lianyungang 222006, China)

Abstract: Loongson 2F handling system in the TLB replacement cost of a larger, existing software technologies from reducing the TLB exception handling heavy loads of time considerations, the use of soft TLB to cache TLB entries, replace the processing Cached improve hit rate and reduce the heavy load exception handling time. To further improve the efficiency of TLB, design multi-page technology to improve TLB coverage, reducing the number of TLB replacement, thereby reducing eth number of exception handling heavy loads, aims to improve system performance.

Key words: TLB; exception handling heavy loads; multi-page technical; Loongson 2F

随着软件规模增大, 内存需求越来越大。现代系统提出虚空间的概念, 软件运行在独立的虚空间中, 被访问时才从物理内存中建立与之对应的空间。TLB(Translate Look-aside Buffer)是加速线性地址向物理地址转换的缓冲器, 即缓存最近使用过的页的物理地址。使用线性地址和 TLB 中的存储项进行比较, 如果相同就直接获得目标页的物理地址, 不需要访问外存, 这样可大大提高地址转换速度。

龙芯 2F 是中科院计算所自主研发的高性能通用 64 位处理器^[1], 在龙芯 2F 的 Linux 系统中, 最占 TLB 替换处理时间的就是访存操作, 减少访存操作就可提高 TLB 使用效率, 从而提高龙芯 2F 的整体系统性能。现有软 TLB 技术可减少 TLB 替换处理时间, 多页技术的采用可以减少 TLB 替换次数, 从而进一步提高 TLB 替换效率, 帮助对龙芯 2F 系统进行优化。

1 龙芯 2F TLB 重载入异常处理

龙芯 2F 系统采用 Debian Linux, 其 TLB 只负责地址

转译, 表项维护由操作系统负责, 所以表页发生修改时 TLB 内容必须被及时更新^[2]。Linux 系统通过 flush_tlb_all、flush_tlb_mm、flush_tlb_page、flush_tlb_range 等函数来操作 TLB 内容。龙芯 2F 为了提高 TLB 重载入异常处理效率, 提供了两个 TLB 专用寄存器 K0 和 K1, 这样在进入重载入异常处理时就不需要对用户寄存器进行保存, 从而减少操作时间。

TLB 重载入异常处理流程为: 进入重载入异常处理后, 首先获取该进程页全局目录首地址, 取得页全局目录和页中间目录内容, 然后获取两个连续页表项用来进行 TLB 替换写入, 处理完毕后返回异常处理现场。

2 软 TLB 处理实现

龙芯 2F 系统为 64 位, 页表为 3 级页表, 且操作地址不连续, 每次访问三级页表内容时, TLB 重载入需要 5 次访存操作, 产生 3 次左右的 Cache 失效, 使得 TLB 重载入异常处理时间大大加长。

《微型机与应用》2011 年第 30 卷第 23 期

技术与方法

Technique and Method

龙芯 2F 系统中设计用软 TLB 技术来减少 TLB 替换处理时间, 软 TLB 技术在内存空间中开辟出一个全局 TLB 缓冲区域, 作为硬件 TLB 的缓存, 在系统进行 TLB 表项替换前, 要先查询该 TLB 缓冲区, 如果所查内容在缓冲区内, 则将内容写入 TLB, 如果查询不到, 则按照原来的 TLB 查询过程进行查询。所以只访问一次缓冲区且访存操作连续的软 TLB 方法产生 Cache 失效的几率被大大降低了^[3]。如果软 TLB 查询失败, 该次查询只会造成系统多作一次访问操作的额外开销, 因此软 TLB 技术减少 TLB 替换处理时间的重点在于提高软 TLB 查询命中率。图 1 为龙芯 2F 软 TLB 重载入异常处理流程。

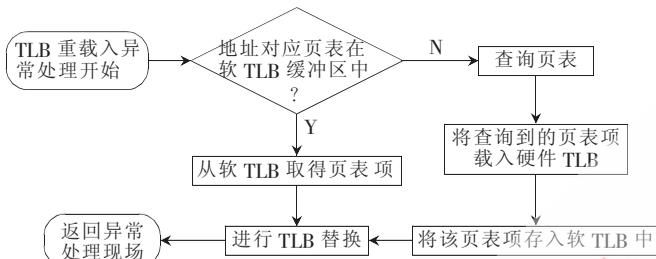


图 1 软 TLB 重载入异常处理流程

系统在修改页表项时, 除修改硬件 TLB 对应项外, 还要查询该页表项是否缓存于软 TLB 中, 若已有缓冲则修改软 TLB 中的页表项, 否则就将该 TLB 表项缓存入软 TLB 中。

3 分页对 Linux 系统的影响

龙芯 2F 采用软硬协同实现内存管理, 硬件完成 TLB 地址转译, 软件完成 TLB 表项替换。为了增大 TLB 的覆盖率, 必须增加每项 TLB 表项的覆盖范围, 系统设计每个表项可以有不同的分页大小, 从 4 KB~16 MB 按 4 倍增长。CPO 寄存器 PageMask 用于记录映射的页的大小, 并且这个记录在写一个新的表项的同时载入 TLB 中^[4]。因此操作系统可以支持不同大小的页表项以适用于不同的目的, 然而在同一运行的时刻只能是固定大小的页。

当 TLB 表项均采用 4 KB 分页大小时, TLB 可为 512 KB 的地址进行转译; 16 MB 分页大小时可为 2 GB 地址进行转译。对于某个运算集合在 4 MB 和 8 MB 之间的程序来说, 如果分页大小为 32 KB, 则在运算集合大于 4 MB 时 TLB 重载入异常处理会大量产生, TLB 替换次数会大量增加。如果分页大小为 64 KB, TLB 就可以覆盖整个 8 MB 运算集合, 会大大减少 TLB 重载入异常的发生。但是分页也不是越大越好, 例如文件系统 I/O 操作与 TLB 无关, 如果用户读取数据小于页大小, 分页过大反而会增加系统 I/O 操作^[4]; 同样, 大的分页会使得系统每个进程所需资源增加, 使系统能支持的进程数减少, 降低系统多进程的性能; 大页还会造成内存使用紧张, 碎片增多, 页的换入换出操作增多, 降低系统内存使用率。

不同分页大小对系统 TLB 重载入异常处理和替换次数有不同影响, 进而对系统性能有不同影响, 所以应

根据不同应用需求采用不同页大小。

4 多页技术的实现

多页技术针对 TLB 每个表项可分配页大小, 根据实际需求按需分配页大小, 该页大小信息保存在页表项中, 用户申请大空间时用大页进行分配, 申请小空间时用大页进行分配, 有利于减少内存碎片。系统最小的页被定为基本页, 其他不同大小的页用多个连续的基本页来表示, 在 TLB 页表项中有页大小的信息项。由于线性区内映射的 TLB 表项中的页属性在页分配时并不分析该页所属线性区内的其他页信息, 所以在建立线性空间时就要将用户所需的页大小信息保存入页表中, 进行页分配时再根据信息进行不同大小的页分配。

多页技术主要通过修改 Linux 系统内存管理子系统实现。线性空间建立主要通过 Linux 内存分配函数 `mmap()` 实现, 应用程序的代码段和数据段均通过 `mmap()` 实现线性空间的建立, 并和文件系统关联, 使得线性空间成为文件内容的映射区域^[5]。要实现多页技术需要对 `mmap()` 进行修改, 为了方便大页的产生, 在查找空闲的线性空间时, 要根据申请空间大小找到满足条件的页大小对齐的线性地址, 如果找不到, 则从大到小进行页对齐。在线性空间分配完成后, 要建立多页属性的页表, 根据传入的线性空间大小, 尽量采用大页, 将分配好的页信息保存至 TLB 页表项的页大小区域内。

完成线性空间建立后, 根据用户需求来决定页大小并将页信息存入页表中, 根据这些信息进行实际页分配。Linux 系统有页缓存页和匿名页两种页面, 为了实现多页技术, 需对这两种页面的分配进行修改。多页技术的修改主要在于对原有页分配函数进行修改替换, 通过对 `filemap_nopage()` 函数的修改实现将分配大页后的文件数据读入到连续页缓存中, 通过对 `do_no_page()` 函数的修改实现页表的建立。具体多页申请分配流程见图 2。

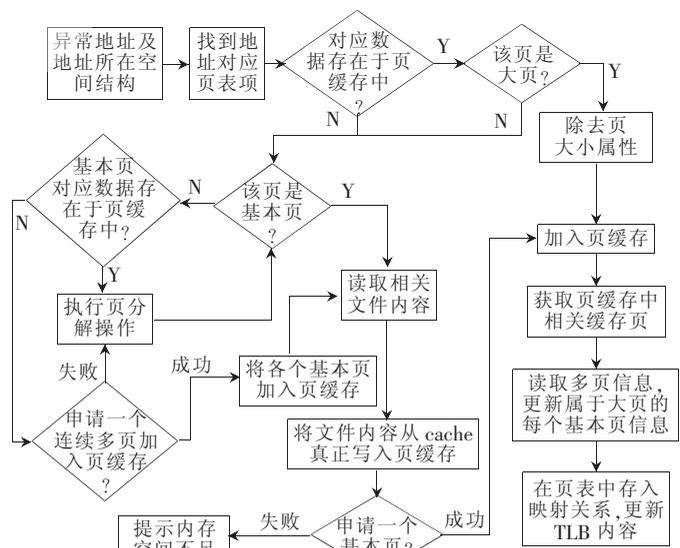


图 2 多页申请分配处理流程

技术与方法 Technique and Method

现有龙芯 Linux 操作系统软件规模日益增大与 TLB 覆盖空间小的矛盾,使得系统在 TLB 替换处理上开销很大,需要在系统内存管理方面进行性能优化^[6]。通过软件 TLB 重载入异常处理实现了 TLB 替换处理中命中率的提高,减少了处理时间。根据用户需求进行分配的多页技术进一步改进了软 TLB 重载入异常处理,提高了 TLB 覆盖率,减少了重载入异常处理次数,能够发挥各种页大小的优势,使得系统性能得到明显提升。

参考文献

- [1] 北京龙芯中科技术服务中心有限公司. 龙芯 2F 处理器用户手册 V1.0.2008.
- [2] 苏波,李凯.龙芯 2F 上的访存优化[J].计算机系统应用, 2010, 19(1): 171-175.

- [3] 许先超. 减少 TLB 失效开销提高 64 位 Linux 系统性能的方法[J].计算机工程, 2006, 32(1): 70-72.
- [4] 孙益辉,陈凯,白英彩.嵌入式操作系统内存管理机制分析及改进 [J]. 计算机应用与软件, 2006, 23 (3): 98-99, 115.
- [5] 谢长生,刘志斌.Linux2.6 内存管理研究[J].计算机应用研究, 2005(3): 58-60.
- [6] 李小群, 孙玉芳.Linux 内存管理机制的分析与研究[J]. 计算机科学, 2002, 29(4), 18-20.

(收稿日期: 2011-07-03)

作者简介:

何炜,男,1980 年生,学士,主要研究方向:嵌入式系统软件。

电子技术应用网
APPLICATION OF ELECTRONIC TECHNIQUE
www.ChinaAET.com