一种基于闪存固态盘的内存交换区空间分配方案

刘青昆,梁 莹,石彦博

(辽宁师范大学 计算机与信息技术学院,辽宁 大连 116081)

摘 要:针对闪存固态盘语义缺失特性,提出了一种带有语义感知的交换区空间分配方案,减少了交换系统中 Trim 命令的使用次数,进而减少内存交换系统中的时间开销,提高程序运行的性能。

关键词:交换系统;闪存固态盘;语义缺失;Trim 命令;空间分配

中图分类号:TP302.7 **文献标识码**:A **文章编号**:1674-7720(2013)13-0070-03

A flash-based solid-state disk memory swap space allocation scheme

Liu Qingkun, Liang Ying, Shi Yanbo

(College of Computer and Information Technology, Liaoning Normal University, Dalian 116081, China)

Abstract: The paper puts forward a kind of swap space allocation scheme with semantic perception, and reduces Trim commands in swap system, thereby reducing the time overhead of memory swap system and improving the performance of applications.

Key words: swap system; flash solid-state disk; semantic gap; Trim command; space allocation

闪存作为一种新型非易失性存储介质,以其低延迟、低功耗、轻重量和抗震性好等优点,在便携式设备和嵌入式系统上得到了广泛的应用[1]。近年来,随着闪存固态盘容量增加和价格下降,其应用逐步走向个人计算机和企业服务器市场[2-4]。在现代计算机系统中,随着应用程序规模的扩大,对内存占用需求的快速增大,如何扩大系统中可用内存空间正受到极大的关注,其中在闪存固态盘上建立交换区已成为研究热点。

本文基于闪存固态盘的自身特点和内存交换系统的具体应用场景,提出一种带有语义感知的内存交换区空间分配方案,使交换系统减少 Trim[5] 命令(也称 discard 命令)的使用,进而节省 Trim 命令引入的时间开销,提高系统性能。

1 相关工作

近些年,越来越多的研究关注于使用闪存固态盘作为交换区,以降低内存与交换区的 I/O 延迟,提高系统性能。参考文献[6]在使用闪存固态盘作为交换区的环境下提出压缩交换页面以提高系统性能,延长闪存寿命的方法。参考文献[7]提出一种基于日志方式的换出和块对齐方式换入的交换系统,以便提高闪存的垃圾回收性能和减少闪存擦除操作次数,进而提高系统性能。参考文献[8]中混合使用传统磁盘和闪存固态盘作为内存交换区,充分发挥闪存和硬盘的特性以提供大容量、低价位、高性能的内存空间。然而,这些系统主要着眼于交换页面的数量或交换页面内容的分析,对内存交换系统进行优化,很少关注交换区空间分配的设计。参考文献[9]指出内存交换系统中带有语义传递功能的 Trim 命令为系统引入了较大的时间开销,在此基础上,本文提出一种带有语义感知的交换区空间分配方案,减少 Trim 命令的使用,以提高程序运行性能。

2 以闪存固态盘为交换区的空间分配

2.1 闪存固态盘的语义缺失

闪存固态盘主要由闪存芯片和闪存控制器两部分构成。与传统磁盘相比,闪存介质在性能和硬件属性上有很多优势,同时在实际应用中存在一些缺陷:首先闪存介质重写闪存页前必须擦除闪存页所在的闪存块;其次闪存块擦除次数有限,若擦除操作达到上限,闪存块可能损坏,影响闪存的使用寿命。由此可知,在传统磁盘中应用广泛的原位重写操作不适用闪存。为使闪存固态盘能像磁盘一样普遍地应用到计算机系统中,Intel 提出闪存转换层 FTL(Flash Translation Layer)技术[10],把闪存固态盘模拟成类似磁盘的块设备,封装在闪存控制器中,统一了闪存固态盘与传统磁盘的软件接口。闪存固态盘的软件设计结构如图 1 所示。

图 1 中 FTL 主要包含地址映射、垃圾回收、磨损均衡等模块。地址映射模块中，将软件系统读写请求的逻辑地址通过映射表和闪存固态盘的物理地址一一对应。当系统原位更新某逻辑地址上的数据时，先通过映射表检索逻辑地址对应的物理地址，标记此物理地址的数据无效，并把数据写入闪存固态盘的其他物理位置，在映射表中使逻辑地址与新的物理地址对应，进而避免重写闪存页前擦除闪存块操作。当闪存控制器启动垃圾回收机制，擦除无效数据较多的闪存块，由于闪存块的擦除次数有限，磨损均衡模块会采取均衡策略，使擦除操作尽量均匀地分配在闪存介质上。

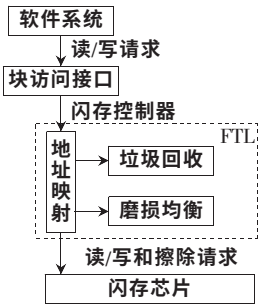

图 1 闪存固态盘的软件设计结构图

由上述分析可知，闪存控制器封装了闪存介质特性，使软件系统能够以相同于磁盘的方法操作闪存固态盘。软件系统向闪存固态盘发出的读写请求经过 FTL 的地址映射后，对闪存介质的操作请求已丧失软件系统操作的语义，因此，闪存固态盘无法感知软件系统操作的相关含义，形成一种语义缺失的现象。例如文件系统对数据的删除操作，在文件系统中的删除操作只标记对应元数据无效，即代表数据已经删除，而闪存固态盘不知道元数据对应的实际数据页面已经无效，直到这些数据页面对应的逻辑页面再次收到写数据请求时，闪存固态盘才能感知这些数据无效，这种因语义缺失导致了垃圾回收操作的延迟，影响闪存固态盘的存储效率。

2.2 Trim 命令的时间消耗评估

闪存固态盘的应用越来越普遍，针对其语义缺失的不足，计算机系统中引入 Trim 命令通知闪存固态盘无效数据的位置，使闪存固态盘感知上层软件系统的语义操作。从 Linux 2.6.29 开始，针对闪存固态盘为交换区，在交换区空间分配方案中引入 Trim 命令，以便闪存固态盘回收交换区中无效页面。其中，交换区空间由 swap_map 计数器数组进行分配管理，每个计数器值对应一个页槽(page slot)使用状态。当内存将暂时不用的非映射页换出到交换区时，需要在 swap_map 中快速检索空闲页槽存储换出页面，页槽为占用状态；当内存需要已换出到交换区中的页面时，再把该页面从交换区中调入内存，此时页槽为空闲状态，交换区空间分配如图 2 所示。

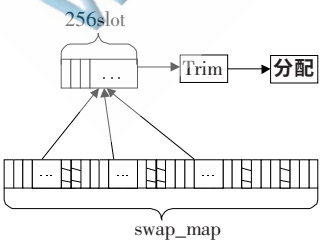

图 2 内存交换区空间分配

图 2 中，空白格代表页槽为空闲状态，阴影小格代表页槽为占用状态，内存交换系统以簇集的形式分配交换区空间，一个空闲簇集包含 256 个连续空闲页槽，当空闲簇集中的空闲页槽全部分配后，从此簇集的最后一个页槽后开始扫描 swap_map，来检索空闲簇集。Trim 命令是在检索到空闲簇集后使用，其作用是通知闪存固态盘 256 个空闲页槽对应的页面无效。其中每个页槽对应 4 KB 的交换区空间，那么 256 个页槽对应 1 MB 的交换区空间，也就是 Trim 命令的作用区间是 1 MB。本文分别对不同作用区间的 Trim 命令进行测试，其时间开销如图 3 所示。

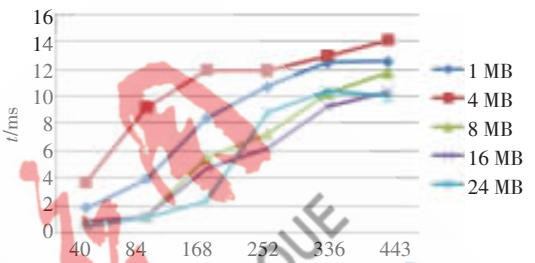

图 3 Trim 命令时间开销

图 3 中横坐标代表换出到闪存固态盘的页面数量，W 代表数量单位为万页，其测试结果是在系统中运行应用程序 ImageMagic 使内存换出大量页面到交换区中，在交换区空间分配时向闪存固态盘发送 Trim 命令，进而跟踪测试不同作用区间的 Trim 命令时间开销。由图 3 可推出两个结论：

(1)Trim 命令平均耗时随着闪存固态盘写入的页面数量增加而增大，当页面数量达到一定值时，Trim 命令平均耗时趋于平稳。整体来看，几种不同作用区间的 Trim 命令平均耗时在 1 ms~14 ms 之间，表明 Trim 命令为系统引入毫秒级别的时间开销；

(2)不同作用区间的 Trim 命令时间开销不同，随着作用区间增大，其平均时间开销减小。

2.3 交换区空间分配方案的设计与分析

内存交换系统使用闪存固态盘作为交换区时，鉴于闪存固态盘语义缺失且语义传递的 Trim 命令耗时较多，本文设计一种带有语义感知的交换区空间分配方案，其交换区空间分配规则如下：

(1)合并 Trim 命令：在 swap_map 中，扫描 16 个连续空闲簇集(1 簇集=256 slot)后，向闪存固态盘发送 Trim 命令，使得 Trim 命令的作用区间为 16 MB。

(2) 语义感知：系统使用完空闲簇集后，扫描 swap_map，从已经写入数据的页槽中检索 1 个空闲簇集，若没有检索到，则从上一个 Trim 命令的作用区间后扫描 swap_map，检索 16 个连续空闲簇集，发送 Trim 命令。

(3)边界对齐：在检索 1 个空闲簇集时，采用边界对齐的方式，使检索到的空闲簇集的边界为 256 的倍数。

第一条规则分析如下：Trim 命令作用区间由 1 MB 增加到 16 MB，首先减少了 Trim 命令的使用次数；其次由 2.2 章节可知，作用区间越大，Trim 命令平均耗时越少，16 MB 作用区间的 Trim 平均时间消耗较少，但是本

方案没有使用作用区间更大的 Trim 命令，是因为 Trim 命令的作用是通知闪存固态盘无效页面的位置，促进闪存固态盘垃圾回收进行闪存块擦除操作，一次发送无效页面的数量较多，可能促使闪存固态盘擦除较多的闪存块，若机器关机或重启，擦除的闪存块会再次被擦除，造成擦除冗余，而闪存块的擦除次数直接影响闪存固态盘的寿命。

第二条规则分析如下：由 2.1 章节论述可知，重写逻辑页面相当于告诉闪存固态盘该逻辑页面对应的物理页面无效，swap_map 数组索引闪存固态盘的逻辑页面，重写数组中的空闲页槽，即是通知闪存固态盘此页槽对应的物理页面无效。该规则利用闪存转换层原位重写操作使闪存固态盘感知无效页面，在内存交换系统中尽量减少 Trim 命令的使用，进而使交换区空间分配带有语义感知功能。

第三条规则分析如下：边界对齐的设计是利用程序运行的局部性原理，内存换出相邻的页面，很可能属于同一进程的页面，当进程被调用时，同时换入这些相邻的页面，使这些页面在交换区中对应的页槽状态变为空闲，边界对齐的分配方法促使这些空闲页槽对应的页面尽可能集中分布同一闪存块中，提高闪存固态盘垃圾回收效率。

3 测试与分析

3.1 测试环境与负载

系统的硬件环境为 Intel(R)Xeon(R)E5420，主频为 2.5 GHz 的四核 CPU，设置物理内存为 1 GB，分别使用传统磁盘和闪存固态盘作为系统的交换区进行测试，设置交换区大小为 2 GB，并选取图像处理 ImageMagick、数据库应用 Postgresql、科学计算 Matlab 以及程序开发 eclipse 四种类型的应用程序对交换区分配方案进行测试分析。其测试负载如下：

(1)ImageMagick：同时放大 10 张 4.5 MB 的图片；

(2)Postgresql：模拟 600 个客户端对数据库 Postgresql 进行压力测试，每个客户端执行 300 个的事务；

(3)Matlab：使用双线性插值法对一张 4.5 MB 的图片放大 4 倍；

(4)eclipse：使用 AES 算法对 260 MB 的压缩包进行加密运算。

以上四个应用程序的负载所需要的内存空间均大于 1 GB，在 1 GB 物理内存的系统下分别单独运行四个应用程序，系统均会产生一定数量的交换操作。在 linux2.6.34.13 系统中实现带有语义感知的交换区分配方案，在内核态统计四种应用程序换出页面频率，在用户态统计程序运行时间，进而测试内存交换区空间分配方案的性能。

3.2 实验结果与分析

在实验中，首先分别使用传统磁盘和闪存固态盘作为交换区，测试四种应用程序，观察不同硬盘对系统性能的影响，如图 4 所示。然后以闪存固态盘为交换区，对比四种应用程序在不同的交换区分配方案下的运行时间，测试交换区分配方案提高系统性能的效果，如图 5 所示。图 4 和图 5 中 Postgresql 应用程序运行时间单位为分钟(min)，其他三个应用程序运行时间单位为秒(s)。

由图 4 可知，使用闪存固态盘作为交换区，相较于传统磁盘，四种典型的应用程序运行性能都有所提高，其中 ImageMagick 性能提高 4 倍，Postgresql 性能提高 4.7%。由图 5 可知，使用本文提出的带有语意感知的交换区分配方案可以进一步提高程序运行性能，ImageMagick、Postgresql、matlab 和 eclipse 在运行的过程中提高性能的

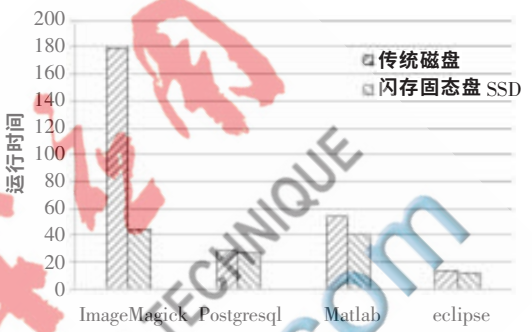

图 4 应用程序在不同硬盘下运行时间对比

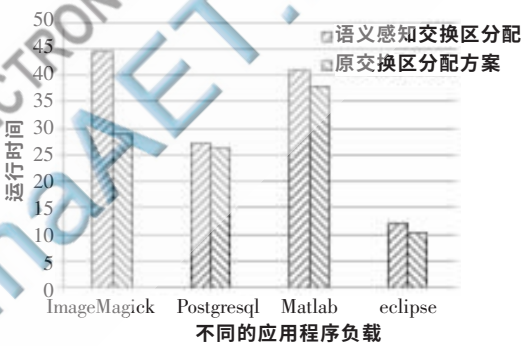

图 5 应用程序在不同分配方案下运行时间对比

比值分别为 34.6%、3.2%、6.6%和 14.2%。对比图 4 和图 5，不难发现在图 4 中性能提高较多的应用程序在图 5 中性能提高比例也较大，说明性能提高的比例与应用程序本身性质有关。为体现交换区分配方案对哪类应用程序有更好的性能提高，本文对四种应用程序特性进行分析，在图 6 中给出了以闪存固态盘为交换区时，四种应用程序运行时对闪存固态盘的写请求频率。

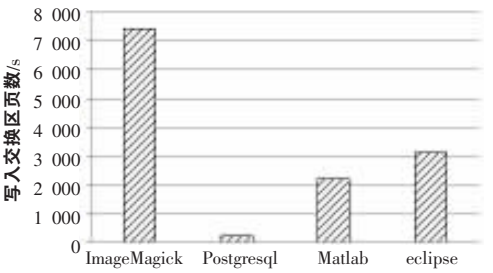

图 6 四种应用程序 I/O 写请求频率对比图

图 6 中四种应用程序运行时对闪存固态盘写请求频率不同，写请求频率高的应用程序在图 5 中性能提高比例也高，写请求频率低的应用程序性能提高也低，结合图 5 和图 6 结果分析可知，对交换区写请求的频率高低直接影响带有语义感知的交换区分配方案的作用效果，写请求频率越高的应用程序性能提高比例越大，即在较短的时间内换出到交换区的页面数量较多的应用程序，其运行性能提高比例较大。

本文主要研究使用固态盘上建立交换区，提出一种语义感知的交换区空间分配方案。通过理论分析和实验证明，该交换区分配方案自带语义理解功能，减少语义传递功能的 Trim 命令使用，进而节约时间开销，提高程序运行性能。在进一步工作中，使用闪存固态盘为交换区，将着重研究通过压缩交换页面或重复数据删除等技术减少闪存固态盘中的写操作，以延长闪存固态盘的寿命，进一步提高系统性能。

参考文献

[1] Sangsung Electronics Co. NAND Flash Memory and Smart Media Data Book [EB/OL]. [2002]. http://www.samsung.com.

[2] GRAY J, FITZGERALD B. Flash disk opportunity for server applications[J]. ACM Queue, 2008, 6(4): 18-23.

[3] LEE S W, MOON B. Design of flash based DBMS: an in page loging approach [C]. In Proc of the ACM SIGMOD, 2007.

[4] LEE S W, MOON B, PARK C, et al. A case for flash memory SSD enterprise database applications [C]. In Proc of the ACMSI MOD, 2008.

[5] WIKIPEDIA. TRIM [EB/OL]. [2013-03-20]. http://en.wikipedia.org/wiki/TRIM.

[6] 顾锋磊. 基于 NAND 的使用压缩缓存策略的交换系统的设计与实现[D]. 上海：上海交通大学，2008.

[7] KO S, JUN S, RYU Y, et al. A new linux swap system for flash memory storage devices[C]. In ICCSA'09, 2008.

[8] LIU K, ZHANG X, DAVIS K, et al. Synergistic coupling of SSD and hard disk for QoS-aware virtual memory[EB/OL]. [2013-02-13]. http://www.cc.gatech.edu/~xczhang/paper/ispass13.pdf.

[9] SAXENA M, SWIFT M M. FlashVM: virtual memory management on flash [C]. Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference. USENIX Association, 2010.

[10] Intel Corporation. Understanding the flash translation layer (FTL)specification [R]. 1998.

（收稿日期：2013-04-11）

作者简介：

刘青昆，男，1971 年生，副教授，硕士生导师，主要研究方向：嵌入式操作系统，并行计算等。

梁莹，女，1988 年生，硕士研究生，主要研究方向：存储系统。