

基于内存网格的磁盘缓存设计与实现

田 甜, 褚 瑞, 谢健聪

(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要:内存对计算机系统的性能具有重要影响,内存网格能够共享跨域的开放网络环境中的内存资源,以磁盘缓存的形式提高系统性能。为实现缓存对应用的透明性,提出了动态修改操作系统内核的二进制代码,实现文件系统读写流程的截获和重定向;并提出了基于内核线程的异步缓存写入方法,提高写缓存的效率。通过原型系统及实验,说明上述方法既不需要修改应用程序,也不需要修改操作系统源代码,并且能充分利用共享的内存资源,提高系统的I/O性能。

关键词:网络内存;内存网格;磁盘缓存;I/O性能

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2011)04-0001-04

Design and Implementation of Disk Cache Based on RAM Grid

TIAN Tian, CHU Rui, XIE Jian-cong

(Department of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: The performance of computer system is impacted by memory. RAM grid can share the memory resources distributed in cross-domain open network environment, and improve the system performance by disk cache. In order to be transparent to the applications, a method of dynamical modifying the binary code of operating system kernel is proposed, which can intercept and redirect the file system read/write routine. A kernel thread-based asynchronous cache writing method is also proposed to improve the effectiveness of cache writing. The prototype system and experiment shows that these methods can take full advantage of the shared memory resources, to improve the system I/O performance, without modifying the source code of the application and the operating system.

Key words: network memory; RAM grid; disk cache; I/O performance

0 引言

在计算机系统中,内存是最重要的资源之一,其容量直接影响系统的整体性能。特别是当运行使用大量内存的“内存密集型”应用或具有大量磁盘访问的“I/O密集型”应用时,物理内存容量起到了决定性的作用。据我们的实践,某气象模拟程序是典型的内存密集型应用,在同一台工作站上把物理内存从4GB降到512MB时,该程序的运行时间从大约30秒增加到20分钟以上^[1]。此外,在对磁盘的随机访问性能测试的过程中,发现空闲内存分别为500MB和50MB时,测试结果也出现了一个数量级的差异。

内存容量对性能的影响,本质源于内外存之间的性能和容量差距过大,基于磁盘的外存在容量价格比上具有很大优势,但其数据传输速度受到了很大限制,I/O操作的速度远远落后于内存的速度,传统的机械

磁盘逐渐成为了计算机系统中性能提升的瓶颈。因此,学术界曾出现过关于“网络内存”的研究,其主要思想是通过共享集群中各节点的内存,构成一种大容量(与内存相比)、高性能(与磁盘相比)、低成本(与购置物理设备相比)的缓存设备,填补内存和磁盘之间的差异,提高内存密集型或I/O密集型应用的性能。但是,网络内存往往需要修改操作系统源码,或者重写应用,其使用价值受到了一定的限制。

在过去的工作中,把网络内存的思想扩展到跨域的开放网络环境中,提出了内存网格的概念^[1];针对开放网络环境中资源的动态性等特点,研究了内存网格的资源管理、性能优化等机制;并基于课题组开发的虚拟计算环境基础软件平台^[2],实现了内存网格的原型系统。然而,对于已有的应用来说,如何在不修改其程序代码的情况下,能够使用到内存网格所提供的跨节点内存共享的功能,仍是一个值得研究的问题。

文中主要提出一种动态修改操作系统内核代码,截获应用程序对文件系统的访问,并按照一定策略将其缓存到内存网格中的方法。

这一方法对于内存网格的推广和应用具有重要的意义。

收稿日期:2010-08-05;修回日期:2010-11-20

基金项目:国家自然科学基金(61003076);国家重点基础研究发展计划(973)(2005CB321801)

作者简介:田 甜(1982-),女,陕西西安人,硕士,研究实习员,研究方向为分布式计算。

1 相关工作

网络内存的研究是从 20 世纪 90 年代开始的。它试图有效利用集群范围内的空闲内存,构造一种性能、容量均处于本地内存和磁盘之间的存储设备,以填补本地内存和磁盘之间的差异。常见的网络内存系统主要应用于两个方面,即内存换页和磁盘缓存,前者主要解决内存密集型应用中换页的性能问题^[3],后者主要解决 I/O 密集型应用中缓存的容量问题^[4]。两者在本质上是一致的,即如何把多个节点的内存共享使用,并把应用程序对本地磁盘的访问重定向到其他节点的内存中。

为实现这样的重定向机制,不同的研究工作提出了不同的方案。

Dodo 是一种用于内存换页的网络内存系统^[5],它通过用户态函数库,提供了五个主要的 API,应用需要有选择地把一部分数据写入到共享内存中,并在需要时再取回。为使用 Dodo 提供的共享内存,需要改写应用程序的源代码,并显式地使用其函数库。这一方案的实现复杂度较低,但对开发人员的要求较高,开发负担较重。而且,很多情况下应用程序的源代码无法获得或者无法修改,Dodo 的作用将完全无法发挥。

全局内存服务(GMS)也是一种经典的网络内存系统^[3]。与 Dodo 最大的区别在于,GMS 通过在 DEC Unix 操作系统上修改内存管理子系统的源代码,将内存换页重定向到其他节点的内存中,实现了对应用的透明性。显然,采用这种方案,虽然应用程序无需修改,但修改操作系统内核的源代码,会给系统的可靠性和安全性都带来了隐患,何况很多操作系统都不是开源的。此外,当操作系统内核需要升级时,这种方案也带来了可移植性的问题。

网络内存盘(NRD)则通过实现操作系统内核驱动模块,在系统中构建一个高性能的虚拟盘^[6],其实际的存储介质是跨节点的内存资源。应用程序可以在基本不修改的情况下,直接采用传统的文件系统接口,把一部分临时数据写入虚拟盘,以提高整体性能。Anemone 则更巧妙地把操作系统的内存交换文件设在虚拟盘上^[7,8],使内存密集型应用不需要修改,即可直接实现更高性能的内存换页。NRD 和 Anemone 的优点在于对应用程序和操作系统的修改都很小,可移植性好,但仅适合部分特定应用,而难以实现磁盘缓存的功能。

通过对已有工作的归纳和分析,文中主要提出动态修改操作系统内核的二进制代码的方法,实现基于内存网格的磁盘缓存功能。能够在对应用完全透明的前提下,把应用的 I/O 操作缓存到跨节点的内存中,以提高系统性能。

2 设计与实现

在之前的工作中,已经实现了内存网格的原型系统,能够把跨域的多个节点组织起来,构造基于 TCP/IP 协议的内存服务^[1],并支持高效的分布式内存资源查找^[9]。文中将利用内存网格实现磁盘缓存,为此,需要截获应用程序对文件系统的读写操作,一方面把写入内容同时发往内存服务作为缓存,另一方面在读出时,如果内存服务中已有缓存,则优先读取缓存内容,以提高性能。

为了不修改应用程序和操作系统的源代码,实现对文件系统读写操作的截获和缓存,在设计过程中,通过 Linux 操作系统内核驱动模块,采用动态修改二进制代码的方法,截获并修改操作系统中的文件系统读写流程。同时,为了提高 I/O 性能,还使用了基于内核线程的异步写缓存的方式来进行缓存数据的更新操作。虽然目前仅在 Linux 操作系统上实现,但其原理也可应用于其他不开源的操作系统中。下文将对具体原理进行阐述。

2.1 文件系统截获

采用函数截获(API Hooking)的方法来动态改变系统函数的执行流程,函数截获(API Hooking)是一种获取特定代码控制权的基本方法^[10]。具有不破坏磁盘映像、支持动态加载和卸载等优点,多用于对不开源系统的监控、调试、逆向工程和扩展。如监控与内存相关的 API 调用可以有效地捕获内存资源泄漏的问题,或对第三方软件添加额外的前置和后置处理过程,等等。

截获系统(Hook System)通常由两部分组成,即截获服务器(Hook Server)和截获驱动(Hook Driver)。截获驱动主要完成实际的截获功能,截获服务器则负责将截获驱动在特定时刻注入(Inject)到截获目标的内存地址空间,并选择性地接收和修改来自截获驱动的信息。根据实现层面的不同,函数截获还可分为应用级和系统级两种。通常需要依据函数截获的目标和使用环境来决定具体采用何种截获方法。

为了构造基于内存网格的磁盘缓存,显然需要对系统中所有应用的 I/O 操作进行截获,因此需要采用系统级的截获方案,通过编写内核模块,直接注入 Linux 内核的内存地址空间,并在内核模块加载的时候,采用动态修改二进制代码的方法,实现截获驱动的主要功能,截获的目标是内核中涉及文件系统读写的函数。具体的实现流程如下:

- 1) 取得文件系统读写的原始函数地址 A;
- 2) 申请代码缓冲区,并将原始函数当前 n 条指令拷贝到代码缓冲区;
- 3) 保存第 $n + 1$ 条指令的地址 B;

4) 在代码缓冲区的结尾位置构造一条跳转指令, 跳转地址为 B;

5) 构造重定向文件系统读写的代码, 设其地址为 C。其中包含对代码缓冲区的调用;

6) 将 A 处原有代码修改为一条间接跳转指令, 跳转地址为 C;

7) 当客户端对文件系统读写时, 将首先跳转到重定向读写的代码中, 并在此过程中包含了完整的对原始文件系统读写函数的调用。

通过上述动态修改的过程, 一旦发生文件系统读写操作, 函数执行流程会直接跳转到文件系统读写重定向过程中, 再通过散列表和 LRU 链等数据结构, 将写操作同时写入本地文件系统和内存网格, 对于读操作, 则快速判断是否在内存网格中已有缓存, 如果有, 则从缓存中读入, 否则从本地文件系统中读入。具体的散列算法及数据一致性问题, 由于不是文中研究的重点, 这里不再赘述。

2.2 异步缓存写入

为了不影响系统效率, 采用了异步写入的方法。由于缓存数据的写操作发生在文件系统的写入函数中, 而文件系统的写入函数运行在核心态, 所以异步缓存写入也选择在核心态实现。并且, 为了使写入过程不对原来的系统执行流程造成较大的性能损失, 通过内核线程来实现异步的写入过程。

传统的 UNIX 操作系统将某些关键任务委托给周期性执行的进程, 包括刷新缓存、交换页帧、维护网络连接等, 但开销太大, 实时性也不够好。现代操作系统多采用内核线程的方式完成这些任务^[11, 12], 由于内核线程运行在核心态, 可以直接调用内核函数, 或直接访问内核的地址空间, 避免了从用户态向核心态切换的过程中所带来的开销, 因此具有较高的效率。

设立了一个环形缓冲区, 以记录文件系统写入操作中所产生的需要缓存的数据, 并使用一个内核线程, 在系统不繁忙的时候, 将已产生的缓存数据批量地写入内存网格, 如图 1 所示。在图 1 中, 环形缓冲区被划分为 N 个块, 一个块除了需要记录被缓存的数据之外, 还需要记录必要的元信息, 包括用于散列定位的块设备号和块号、块大小、块是否为脏等等。

基于图 1 所示的数据结构, 具体的写入过程如下:

1) 截获文件系统的写入函数, 将有必要缓存的数据写入环形缓冲区写指针的位置;

2) 如缓冲区已满, 唤醒写数据的内核线程, 并放弃缓存;

3) 内核线程的调度优先级较低, 仅当系统空闲时才获得 CPU;

4) 内核线程从读指针的位置开始检查每个块是

否已置脏标志;

5) 锁定环形缓冲区, 将置脏标志的缓存数据写入内存网格;

6) 内核线程解锁环形缓冲区, 让出 CPU, 进入睡眠。

通过上述过程, 缓存数据总是在系统空闲时被写入内存网格, 确保了系统的原始写入性能几乎不受到影响。

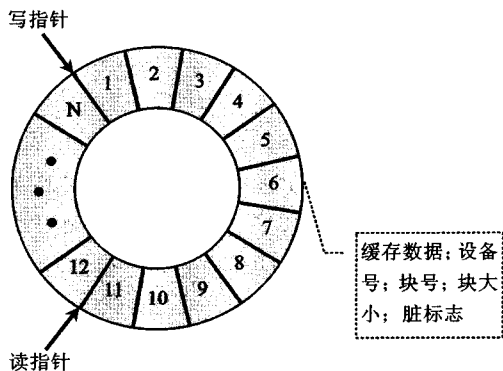


图1 异步缓存写入过程中的关键数据结构

3 实验结果

在国家 973 计划项目的支持下, 本课题组已建立了开放网络的仿真实验平台, 包含 3 个管理域和 1 个网络损伤模拟器, 每个域内包括路由器、硬件防火墙以及 20 ~ 30 个专用计算或存储节点, 域内节点间通过千兆网络相连, 各个管理域接入网络损伤模拟器, 以仿真广域网络环境。在上述实验环境的两个域中选取了 32 个节点进行实验, 各节点均采用 Linux 2.6.18 版本内核。

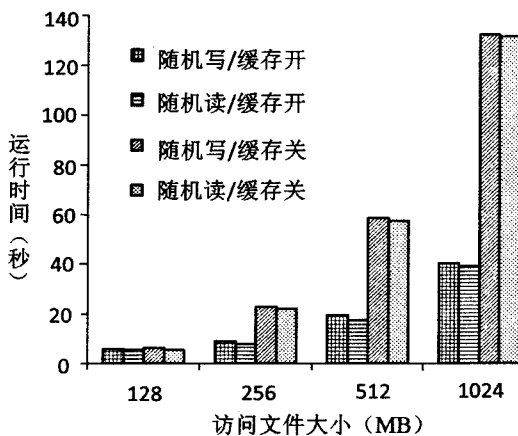


图2 随机读写访问测试结果

编写了具有 I/O 密集型特点的测试程序, 对某个大文件先后进行粒度为 32KB 的随机写和随机读的操, 并测试其完成时间。为便于比较, 对文件大小为 128MB、256MB、512MB、1024MB 的情况分别进行了测试, 并测试了打开与关闭内存网格磁盘缓存的情况。

结果如图 2 所示,当关闭内存网格磁盘缓存时,虽然操作系统所提供的本地缓存仍然发挥作用,但由于容量限制,当访问的文件较大时,性能受到了一定影响。而内存网格提供了额外的缓存空间,对本地缓存形成了很好的补充,使系统性能得到了一定的提高。并且,测试程序不需进行任何修改或重编译,即可使用到内存网格磁盘缓存的功能,进一步增强了本系统的实用性。

4 结束语

内存网格能够共享开放网络环境中的内存资源,并以磁盘缓存的方式提高 I/O 密集型应用的性能。为实现缓存对应用的透明性,文中基于已有研究及原型系统,对采用内存网格实现磁盘缓存的关键技术进行了研究和实现,提出了基于文件系统截获的动态读写重定向方法,以及基于内核线程的异步缓存写入方法。实验证明,文中提出的方法对于内存网格的推广应用具有重要的作用。

参考文献:

- [1] 褚瑞,肖依,卢锡城. 一种基于内存服务的内存共享网络系统[J]. 计算机学报, 2006, 29(7): 1225-1233.
- [2] 卢锡城,王怀民,王戟. 虚拟计算环境 iVCE: 概念与体系结构[J]. 中国科学 E 辑, 2006, 36(10): 1081-1099.
- [3] Feeley M J, Morgan W E, Pighin F H, et al. Implementing Global Memory Management in a Workstation Cluster[C]//Symposium on Operating Systems Principles. Copper Mountain Resort, Colorado: [s. n.], 1995.
- [4] Dahlin M D, Wang R Y, Anderson T E, et al. Cooperative Caching: Using Remote Client Memory to Improve File System Performance[C]//the First Symposium on Operating Systems Design and Implementation. Monterey, Calif: [s. n.], 1994.
- [5] Acharya A, Setia S. Availability and Utility of Idle Memory in Workstation Clusters[J]. ACM SIGMETRICS - Performance Evaluation Review, 1999, 27(1): 35-46.
- [6] Flouris M D, Markatos E P. The Network RamDisk: Using Remote Memory on Heterogeneous NOWs[J]. Cluster Computing, 1999, 2(4): 281-293.
- [7] Hines M, Wang J, Gopalan K. Distributed Anemone: Transparent Low-Latency Access to Remote Memory in Commodity Clusters[C]//International Conference on High-Performance Computing. Bangalore, India: [s. n.], 2006.
- [8] Hines M, Lewandowski M, Wang J, et al. Anemone: Transparently Harnessing Cluster-Wide Memory[C]//International Symposium on Performance Evaluation of Computer and Telecommunication Systems. Calgary, Alberta, Canada: [s. n.], 2006.
- [9] 褚瑞,卢锡城,肖依. 一种基于聚类的虚拟计算环境资源聚合方法[J]. 软件学报, 2007, 18(8): 1858-1869.
- [10] 王全民,周清,刘宇明,等. 文件透明加密技术研究[J]. 计算机技术与发展, 2010, 20(3): 147-150.
- [11] 毛德操,胡希明. Linux 内核源代码情景分析[M]. 杭州: 浙江大学出版社, 2001.
- [12] 肖竞华,陈岚. Linux 内存管理实现的分析与研究[J]. 计算机技术与发展, 2007, 17(2): 187-189.

《计算机技术与发展》投稿要求

(1) 新投稿可通过 Email 发至本刊电子信箱: ctad@vip.163.com。投稿前请作者自审一遍,论文要求主题突出、用语规范、层次清楚、结构严谨、文字精练、文理通顺、逻辑性强。

(2) 论文题目不超过 20 个汉字。

(3) 作者姓名及作者所在单位部门、城市、邮政编码(多位作者不在同一单位应分别开列)。

(4) 摘要须从目的、方法、结果、结论 4 个方面阐述,200 字以上。

(5) 关键词 3~8 个为宜。

(2)~(5) 项内容必须中、英文具备。

(6) 作者简介: 姓名、出生年、性别、学位、研究方向;

导师简介: 姓名、职称、研究方向。

(7) 作者在投稿时须注明是否是中国计算机学会(CCF)会员(高级会员、普通会员、学生会会员)。若是会员,请注明会员号(凡第一作者为 CCF 会员/高级会员/学生会会员者,将享受 85 折的版面费优惠。)

(8) 投稿时请写明详细通信地址、邮政编码、联系电话、Email 信箱等各项必备内容。收到稿件经初审通过后,30 天内以电子邮件的方式通知作者处理意见。稿件刊登后赠送样刊 2 本。