基于语义信息的存储能效的研究

尤红桃 张延园 林 奕 刘 胜

(西北工业大学计算机学院 西安 710129)

摘 要 随着数字化信息爆炸性的增长,存储技术成为IT业发展的新动力。存储系统规模的不断扩大,使能效问题越来越突出,主要表现为增加了系统运行维护和冷却的成本、降低了系统的可靠性和扩展性、加剧了存储系统周围环境的污染,因此研究存储能效问题具有较大的经济价值和实用意义。阐述了存储系统中磁盘能效的研究进展和现状,并从语义信息出发,设计与实现了基于语义信息的驱动程序。实验表明该驱动程序有效地降低了磁盘能耗,提高了存储系统I/O性能,优化了存储能效。

关键词 存储,能耗,能效,语义信息,I/O性能

中图法分类号 TP319 **文献标识码** A

Based on the Semantic Information of the Stored Energy Efficiency Research

YOU Hong-tao ZHANG Yan-yuan LIN Yi LIU Sheng

(College of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract With the digital information explosive growth, storage technology has become a new impetus to the development of IT industry, constantly expanding the size of the storage system, energy efficiency problem is becoming more and more serious, main show is increase of system operation and maintenance cost, reduces the system operation reliability and expansibility, exacerbated by the storage system to environmental pollution, so the study of stored energy efficiency has great economic value and practical significance. The article elaborated the storage system disk energy current situation and research progress, and from the semantic information of design and realization, based on the semantic information of the driver. Experimental results show that the driver can effectively reduced the energy consumption of the disk storage system, improved the I/O performance, optimized the energy efficiency of storage.

Keywords Storage, Energy consumption, Energy efficiency, Semantic information, I/O performance

1 引言

随着IT的发展、纳米技术的大规模应用,数据呈爆炸性的增长,这对存储系统能效问题提出了越来越高的挑战。数据存储已演变成数据中心能量消耗的最大者。基本的解决方法是通过减少和合并来自各方面的I/O请求、提高存储I/O速度、降低单个磁盘的能耗、减少数据中心中的工作磁盘数、使之能够合理地容纳更少的磁盘来满足存储系统能效的要求。

文献[1]通过减少磁头寻道开销来降低能耗,主要方法是为数据创建多个副本并将其存储在文件系统的空闲块上,通过I/O调度的方法使用户请求尽可能地顺序访问磁盘上的数据,从而既提高了用户性能又有效地降低了能耗。文献[2]通过延长磁盘处于空闲状态的周期,对存储系统中的cache替换策略、数据分布、磁盘调度进行研究,提出一种基于成本的cache替换算法,该算法提高了存储系统的存储速度,降低了存储能耗。文献[3]提出动态转速磁盘的概念,即将磁盘的盘片旋转速度分为多个等级,当系统负载较轻时使磁盘运转在低速旋转状态,当系统负载变重时,将磁盘相应调整为高速旋转状态,从而有效地降低了存储能耗。而基于语义的存储利用上层文件系统提供的元数据信息和语义属性来探索文件和数据块之间的相关性,以提高预取的准确性[4],同时凭借着各层之间信息的动态交互,对热点数据的实时性把握,能够有效地解决存储能效问题。

2 相关知识

2.1 语义信息

语义信息(semantic information)是指任何有含义的语言、文字、数据、符号等提供的信息,而信息就是我们以前不知道的事物特定的状态,比如天气预报、命题或描述语句、预言、科学理论等提供的信息。在存储系统中语义信息主要是指文件系统或数据库系统在其生命周期内产生的一系列信息所表现出来的含义[5,6]。论文所述的语义信息是指The Second Extended File System(Ext2)文件系统的语义信息。

2.2 Linux Ext2数据布局

Ext2文件系统是Linux系统中的标准文件系统,通过对Minix文件系统进行扩展得到,负责存储和组织文件,其存取文件的性能良好,是一个基于磁盘的文件系统。Ext2文件系

本文受国家科技支撑项目(2011BAH04B05)资助。

尤红桃(1987—),男,硕士生,主要研究领域为网络存储;张延园(1954—),男,教授,硕士生导师,主要研究领域为软件工程、网络软件、存储网络;林 奕(1976—),男,副教授,硕士生导师,主要研究领域为网络存储、实时存储;刘 胜(1987—),男,硕士生,主要研究领域为网络存储。

统将整个分区划分成若干个同样大小的块组(Block Group)，各块组由以下几个部分组成。

超级块(Super Block)描述整个分区的文件系统信息。

组描述符表(GDT，Group Descriptor Table)存储一个组块的描述信息。

块位图(Block Bitmap)描述整个块组中哪些块已用，哪些块空闲，bit 为 1 表示该块已用，bit 为 0 表示该块空闲可用。

inode 位图(inode Bitmap)描述整个块组中哪些 inode 已用，哪些 inode 空闲。

inode 表(inode Table)包含一个块组中的所有文件的 inode 信息。

数据块(Data Block)用于存储数据。对于常规文件，数据存储在数据块中。对于目录，该目录下的所有文件名和目录名存储在数据块中。对于符号链接，如果目标路径名较短则直接保存在 inode 中以便于查找，如果目标路径名较长则分配一个数据块来保存。对于设备文件、FIFO、socket 等特殊文件没有数据块，设备文件的主设备号和从设备号保存在 inode 中。

Linux Ext2 磁盘数据布局如图 1 所示。

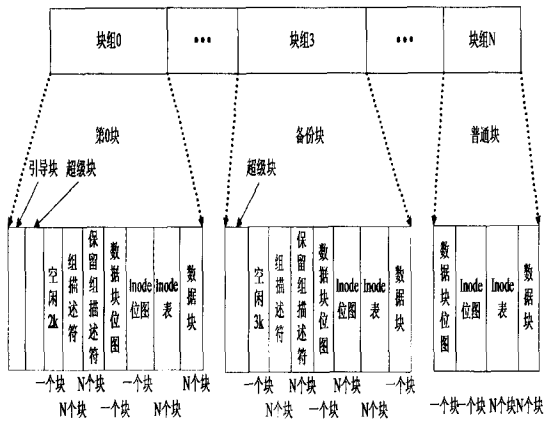

图 1 Ext2 磁盘数据布局

2.3 热点数据

热点是普遍存在的现象，在现实生活中，很多解决问题的方法和时机都是围绕着热点进行的。存储系统中热点主要是热点的数据，它划分为两大类：永久性热点数据(例如天气情况)和阶段性热点数据。其中阶段性热点还可以继续分为两个小类：周期性热点(伦敦奥运会)和突发性热点(日本海啸)[7]。利用好热点数据可以有效地提高存储系统的性能，比如复制热点数据是一种经济有效的方法。本文对热点数据的研究主要是针对单个存储磁盘。

3 基于语义信息的存储能效

根据 sun 公司提出的能效计算公式 Swap=性能/(占用空间×功耗)可以从提高系统 I/O 性能和降低存储能耗两个方面解决能效问题。由于磁盘在空闲时刻仍然会消耗大量的能量，文献[8]采用了固定超时策略让磁盘进入低功耗模式甚至关闭磁盘，即当磁盘空闲的时间超过一定的临界点时，就立刻关闭电源，以减少能量的消耗；当有数据访问到达时，就重新开启磁盘为数据访问服务。文献[9]提出 Hibernator 模型，在基于动态转速磁盘模型的基础上，Hibernator 将数据迁移到合适转速的磁盘上，从而在保证满足性能要求的前提下达到节能的目的。然而上述两个方法在解决存储能耗的同时存在着一定的缺陷，一方面，硬件实现难度大，例如无法准确地确定固定超时策略的时间点。软件实现起来比较复杂，要综合考虑到磁盘调度、数据布局等一系列问题。另一方面没有充分利用存储子系统自身所提供的强大的处理能力。本文解决能效问题的思路是：通过获得上层文件系统的语义信息，推断出热点数据块，对其进行有效的数据布局，从而降低存储能耗，提高存储系统 I/O 性能。

3.1 存储语义信息及其获取方法

语义信息包含静态信息和动态信息。静态语义信息指文件系统/数据库系统本身所固有的信息，如 inode，superblock，bitmap 等。动态语义信息指文件系统/数据库系统在其生命周期内对文件所含的数据块进行操作时产生的属性的变化。

3.1.1 语义信息的获取方法

3.1.1.1 分类

分类主要是为了划分数据块的类型，包括直接分类和间接分类两种方法[10]。直接分类是最简单也是最有效的识别数据块类型的方法，通过执行一个简单的边界检查，计算块范围属于哪个特定的块。例如在文件系统中 inode 号(i_ino)是统一编址的，因此可以根据 inode 号来确定这个 inode 所在的组号以及组内的偏移，进而找到这个 inode。根据图 1，inode 结点的地址(字节偏移)按如下的方法计算：

$$inode\_address=((\frac{inode\_no}{inode\_per\_group})\times block\_perg+(1+n+n+1+1)\times is\_backup\_group)*block\_size+(inode\_no\%\ inode\_per\_group)*inode\_size$$

$inode\_per\_group$ 为每个数据组块含有多少个结点，$block\_perg$ 为每个组块的块的个数，$inode\_no$ 为索引节点号，$1+n+n+1+1$ 表示 1 个超级块，n 个组描述符表，n 个保留组描述符表，1 个数据块位图，1 个索引结点位图。对于 superblock，bitmaps 等同样可以通过计算得到。由于数据块的类型在文件系统执行期间动态地变化，利用直接分类无法准确判断它的类型，间接分类能准确地解决数据块分类问题。判断一个数据块是属于文件还是目录是间接分类的主要内容。在捕获的所有来往的 inode 数据流中，当观察到数据块是目录 inode 时，把它所对应的数据块号码插入到名为 dir_blocks 的哈希表中，当观察到数据块位图从 1 到 0 的变化时，说明它对应的数据块被释放，从 dir_blocks 中移除此数据块号。因此判断一个数据块是属于文件还是属于目录，直接扫描 dir_blocks 哈希表即可。

3.1.1.2 联系

分类只能简单地识别出数据块的类型，但各个数据块之间的语义关系还不太明确，通过联系可以为各个数据块之间建立语义关系，例如通过对目录和它所包含的 inode 进行联系，可以推断出某个给定的 inode 的路径。当然最重要的是将数据块和它对应的 inode 建立联系[10,11]。通过联系可以对普通数据和元数据进行区分，对于元数据，驱动程序根据磁盘分区的情况，将其复制 N 份(其中 N 是磁盘分区的个数)，对于普通数据，如果文件系统访问到它所对应的 inode，则驱动程序将它取到磁盘缓冲区，以提高磁盘的读写性能。

3.1.1.3 操作分析

分类和联系是提供识别磁盘块的类型和建立磁盘块语义的有效方式。而操作推断是理解数据块语义信息变化的关键[10,11]，论文主要分析文件的建立操作和删除操作。确认文件的建立和删除有两个步骤：第一步实际检测到文件的建立

和删除。例如当一个 inode 块被修改的时候，驱动程序分析它是否建立或删除，一个合法的 inode 块有一个非 0 的修改时间和 0 的删除时间，当上述时间发生变化时，证明 inode 对应的文件被建立，否则文件被删除；第二步是根据 inode 位图发生的变化进行分析。例如当 inode 位图中的 1 个 bit 从 0 变成 1 或从 1 变成 0 时，说明 inode 对应的文件被建立或被删除。

3.2 提高存储能效的具体步骤

3.2.1 热点数据块的确定

热点数据主要有两个显著的特点：一是大部分的数据访问都是读；二是数据拥有较高的访问频率[7]。论文在 inode 结构中加入一个记录着一定时间内访问文件次数的字段 inode_tmp_count，通过捕获文件访问时间 i_atime 的变化来实现 inode_tmp_count 自增操作。当 inode_tmp_count 在一个给定的时间内达到了一个预先定义的数值时（通过宏定义），观察记录这些数据块的数据位图，读出数据位图的数据值，并且和 1 进行比较（因为位图字段显示 1，说明它对应的数据块上面存在着数据），如果是 1，将它确定为热点数据块。同一个 inode 对应的一个或者多个热点数据块隶属于同一个热点文件（因为一个文件只有一个 inode）。

3.2.2 优化数据布局

文件的物理结构主要有 3 种：顺序结构，索引结构，链接结构。Ext2 文件系统中的文件在磁盘上的物理结构属于链接结构，它的优点是可以动态地增加、删除，但是要完整地遍历一个文件所包含的数据块，需要多次移动磁头，从而浪费了大量的能耗。论文根据两种理论来优化数据布局，首先是时间和空间的局部性原理，主要是指一个文件被访问后不久将会被又一次访问，文件包含的前一个数据块内容被访问，下一个数据块内容即将被访问。其次是磁盘缓存的速度比磁盘速度快，但是磁盘缓存上未必都是热点数据块，因此将包含热点数据的数据块复制到磁盘缓存中，同时判断磁盘缓存是否已满（磁盘缓存比较小，一般为 4～16M）以及是否可以包含一个文件的所有数据块，如果磁盘缓存已满，将剩下的热点数据块复制到磁盘读写头所在的磁道上。如果缓冲区的大小大于文件的大小，则根据 i_blocks 字段的值判断是否已经将同一个文件中的所有数据块复制到缓冲区。通过优化数据布局，提高了系统的 I/O 性能、降低了存储能耗。

3.2.3 调整磁盘状态

磁盘的状态大致可以分为 4 个主要状态：活动状态、空闲状态、待机状态、停机状态。当磁盘处于活动状态时，磁盘面高速旋转，磁盘读写头寻道，传输着数据，此时磁盘的能耗最大；当磁盘处于空闲状态时，盘面保持旋转，其它部分电子器件处于关闭状态，此时相对于活动状态消耗的能量少；当磁盘处于待机状态，电子器件关闭，磁盘盘面不再旋转，此时消耗的能量非常低；当磁盘处于停机状态时，与磁盘相关的电子器件全部关闭，此时不消耗能量。本文采用了类似于文献[4]所述的技术。通过观察 inode 表可知，如果在某一个时间段内（大于 4 秒）上层对磁盘没有任何操作，则磁盘进入低功耗模式甚至关闭磁盘。由于论文是对单个磁盘进行能效研究，因此并不存在真正意义上的关闭磁盘。

4 驱动程序的实现及实验

4.1 驱动程序的实现

论文在 Linux 2.6 内核基础上设计了一个基于语义信息的驱动程序，该驱动程序以模块的方式动态地加载到内核，作为内核的一部分与内核同步工作。系统磁盘 WDC WD800JD-60LUA0，文件系统为 Ext2。

4.2 实验与结果分析

4.2.1 数据块读写实验与分析

磁盘的主要功能是提供给用户读写和存储用户数据。读写速度是磁盘系统历来研究的重点。从图 2 和图 3 可以看出基于语义信息的读写速度明显高于正常磁盘的读写速度。这种情况在多个多媒体请求下体现得尤为明显，主要原因是多媒体请求基本上属于顺序读写，而本文采用的预取策略和数据智能分布的方法恰好满足要求。

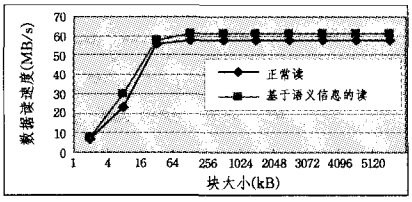

图 2 磁盘读性能曲线

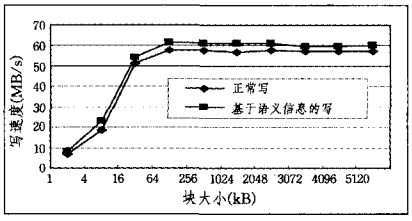

图 3 磁盘写性能曲线

4.2.2 平均响应时间实验与分析

响应时间是指应用程序从发出请求到存储系统做出反应（响应）的时间，对于实时性要求比较高的应用来说这是一个关键的指标。从图 4 中可以看出，基于语义信息的平均响应时间明显低于正常磁盘的响应时间，原因是驱动程序采用了预取策略和优化数据分布的方法。

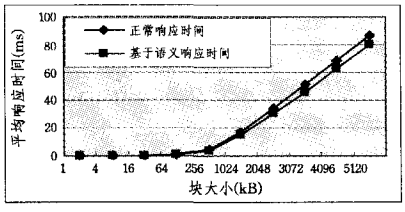

图 4 平均响应时间曲线

4.2.3 单个磁盘能耗实验与分析

磁盘能耗已成为存储系统研究的热点，从图 5 可以看出基于语义信息的磁盘能耗少于正常情况下的磁盘能耗，主要原因是提高了系统性能，动态地调整了磁盘状态。

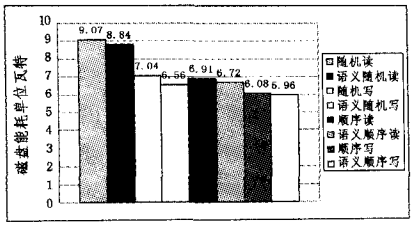

图 5 单个磁盘能耗柱线图

结束语 磁盘的机械特性决定了磁盘在存储系统中是消耗能量的主要来源。而基于语义的存储凭借着各层之间信息

（下转第 148 页）

过该值，合法的商业邮件群发数量也应该参考该值。

结束语 本文分析了服从幂律分布的网络的3个特性，提出了一种节点分类的统计学方法。社会网络中存在幂律分布，其中的节点可以被阈值 T 分成两类。T 值的确定与参数 γ 以及观察者对小概率事件概率 α 的取值相关，而与网络中节点规模 N 无关。网络中大量的异常行为会影响应有的分布规律，导致超过 T 值的节点数远远高于正常值。通过程序分析邮件服务商提供的样本数据验证了该方法的有效性。

在如何有效地建立网络模型方面仍有改进的空间。另外，僵尸网络与垃圾邮件之间有很密切的联系，利用超图理论通过检出的垃圾邮件发送节点寻找僵尸网络也将是一个可行的研究方向。

参 考 文 献

[1] Kleinberg J. The small-world phenomenon: An algorithmic perspective[C]// ACM Symposium on Theory of Computing. 2000, 32

[2] Newman M E J. Models of the small world[J]. Journal of Statistical Physics, 2000, 101: 819-841

[3] Watts D J. The "New" Science of Networks[J]. Annual Review of sociology, 2004, 30: 243-270

[4] Watts D J, Strogatz S H. Collective dynamics of 'small-world' networks[J]. Nature, 1998, 393: 440-442

[5] Newman M E J, Strogatz S H, Watts D J. Random graphs with arbitrary degree distributions and their applications[J]. Physical Review E, 2001, 64

[6] Clauset A, Shalizi C R, Newman M E J. Power-law distributions in empirical data[J]. SIAM Review, 2009, 51: 661-703

[7] Newman M E J. Power laws, Pareto distributions and Zipf's law [J]. Contemporary Physics, 2005, 46: 323-351

[8] Iversen G R, Gergen M. 统计学[M]. 吴喜之，等译. 北京：高等教育出版社，2002: 235-237

[9] Arbesman S, Kleinberg J, Strogatz S. Superlinear Scaling for Innovation in Cities[J]. Physical Review E, 2009, 79

[10] Cohen R, Havlin S. Scale-Free Networks Are Ultrasmall[J]. Physical Review Letters, 2009, 90

[11] Easley D, Kleinberg J. Networks, Crowds, and Markets: Reasoning About a Highly Connected World[M]. Cambridge University Press, 2010: 63

[12] Symantec Corp. Symantec Announces August 2011 Symantec Intelligence Report[EB/OL]. http://www.symantec.com/about/news/release/article.jsp?prid=20110823_01, 2011-08-23

[13] 张铭峰，李云春，李巍. 垃圾邮件过滤的贝叶斯方法综述[J]. 计算机应用研究，2005(8): 14-19

[14] 王斌，潘文锋. 基于内容的垃圾邮件过滤技术综述[J]. 中文信息学报，2005(8): 1-10

[15] Zhao Y, et al. BotGraph: Large Scale Spamming Botnet Detection[C]// Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (USENIX, Berkeley, CA). 2009: 321-334

(上接第 114 页)

的动态交互、数据的高效管理和组织、上下文随机存储数据一致性等优点，充分利用磁盘并发所带来的性能优势已经在语义网络、数据库等 I/O 密集型的应用程序中得到了广泛的应用。本文通过语义信息对存储能效进行了研究，实验表明基于语义信息的存储优化了系统性能(读写速度和响应时间)、降低了存储能耗，当然论文仅降低了单个磁盘的能耗，提高了单个磁盘的读性能，对于系统各项 I/O 性能的提高有待进一步研究，有关理论和计算结果还需进行实验验证。

参 考 文 献

[1] Huang Hai, Huang Wan-da, Shin G K. FS2: Dynamic Data Replication in Free Disk Space for Improving Disk Performance and Energy Consumption[C]// Proceedings of the 20th ACM Symposium on Operating Systems Principles (SO-SP). NewYork: ACM, 2005: 263-276

[2] 仇德成. 网络存储 cache 替换与磁盘调度算法研究[D]. 兰州：兰州大学，2007

[3] Gurumurthi S, Sivasubramaniam A, Kandemir M, et al. DRPM: Dynamic Speed Control for Power Management in Server Class-Disks[C] // Proceedings of the International Symposium on Computer Architecture(ISCA). New York: ACM, 2003: 169-181

[4] 王娟. 对象存储系统中元数据管理研究[D]. 武汉：华中科技大学，2010

[5] 夏鹏. 文件系统语义分析技术研究[D]. 武汉：华中科技大学，2011

[6] 肖亮. 基于服务质量的对象存储优化研究[D]. 武汉：华中科技大学，2009

[7] 吴晨涛. 对象存储系统中热点数据的研究[D]. 武汉：华中科技大学，2010

[8] Papathanasiou E A, Scott L M. Energy Efficient Prefetching and Caching[C]// Proceedings of the USENIX 2004 Annual Technical Conference(USENIX). Berkeley, CA, USA: USENIX, 2004: 255-268

[9] Zhu Qing-bo, Chen Zhi-feng, Tan Lin, et al. Hibernator: Helping Disk Arrays Sleep through the Winter[C]// Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP). NewYork: ACM, 2005: 177-190

[10] Sivathanu M, Prabhakaran V, Popovici F I. Semantically-Smart Disk Systems[J]. Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST '03), 2003, 33 (4): 73-78

[11] Sivathanu M. Semantically-Smart Disk Systems[D]. NewYork: University of Wisconsin-Madison, 2001