软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(7):1824−1839 [doi: 10.13328/j.cnki.jos.004606]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

Ripple-RAID:一种面向连续数据存储的高效能盘阵*

孙志卓[1,2], 张全新[1,3], 谭毓安[1,3], 李元章[1,3]

[1](北京理工大学 计算机学院,北京 100081)

[2](德州学院 计算机系,山东 德州 253000)

[3](北京理工大学 北京市海量语言信息处理与云计算应用工程技术研究中心,北京 100081)

通讯作者: 张全新, E-mail: zhangqx@bit.edu.cn

摘 要: 视频监控、备份、归档等应用具有独特的负载特性和I/O访问模式,需研究特定的存储节能方法.磁盘阵列的局部并行策略有利于实现该类存储系统的节能,但通常会导致RAID执行小写操作而严重影响性能.为此,提出一种面向该类存储系统的高效能盘阵——Ripple-RAID,采用新的局部并行数据布局,通过综合运用地址转换、异地更新、基于流水技术渐进生成校验、分段数据恢复等策略,在单盘容错条件下,保持了局部并行的节能性,又有效解决了局部并行带来的小写问题.Ripple-RAID 具有突出的性能和节能效率,在 80%顺序写负载情况下,请求长度为512KB时,写性能为S-RAID 5的3.9倍,Hibernator、MAID写性能的1.9倍,PARAID、eRAID 5写性能的0.49倍;而比S-RAID 5节能20%,比Hibernator、MAID节能33%,比eRAID 5节能70%,比PARAID节能72%.

关键词: 高性能;节能;盘阵;视频监控;归档;连续数据存储

中图法分类号: TP333

中文引用格式: 孙志卓,张全新,谭毓安,李元章.Ripple-RAID:一种面向连续数据存储的高效能盘阵.软件学报,2015,26(7):1824−1839. http://www.jos.org.cn/1000-9825/4606.htm

英文引用格式: Sun ZZ, Zhang QX, Tan YA, Li YZ. Ripple-RAID: A high-performance and energy-efficient RAID for continuous data storage. Ruan Jian Xue Bao/Journal of Software, 2015,26(7):1824−1839 (in Chinese). http://www.jos.org.cn/1000-9825/4606. htm

Ripple-RAID: A High-Performance and Energy-Efficient RAID for Continuous Data Storage

SUN Zhi-Zhuo[1,2], ZHANG Quan-Xin[1,3], TAN Yu-An[1,3], LI Yuan-Zhang[1,3]

[1](School of Computer Science, Beijing Institute of Technology, Beijing 100081, China)

[2](Department of Computer, Dezhou University, Dezhou 253000, China)

[3](Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application, Beijing Institute of Technology, Beijing 100081, China)

Abstract: The applications, such as video surveillance, backup and archiving, have inherent workload characteristic and I/O access pattern, and require specialized optimization for storage energy saving. Partial parallelism in RAID is beneficial to storage energy saving, but generally makes RAID perform small writes, which heavily deteriorates the performance. A high-performance and energy-efficient RAID, Ripple-RAID, is proposed for these applications. A new partial-parallel data layout is presented, and by comprehensively employing strategies such as address mapping, out-place update, generating parity data progressively based on pipeline, and segmented data recovery, Ripple-RAID not only obtains energy efficiency of partial parallelism but also eliminates the small writes incurred by partial parallelism while providing single disk fault tolerance. When write workload is 80% sequential and transfer request size is 512KB, the write performance of Ripple-RAID is 3.9 times that of S-RAID 5, 1.9 times that of Hibernator and MAID, and 0.49 times that of PARAID and eRAID 5; meanwhile its energy consumption is 20% less than S-RAID 5, 33% less than Hibernator and MAID, 70% less than eRAID 5, and 72% less than PARAID.

* 基金项目: 国家自然科学基金(61370063); 国家高技术研究发展计划(863)(2013AA01A212)
收稿时间: 2013-04-17; 修改时间: 2013-09-09; 定稿时间: 2014-04-24

Key words: high performance; energy efficiency; RAID; video surveillance; archiving; continuous data storage

视频监控、连续数据保护(continuous data protection,简称 CDP)、虚拟磁带库(virtual tape library,简称 VTL)、备份(backup)、归档(archiving)等应用日益广泛,仅就视频监控系统而言,IMSResearch 报告 2012 年市场总规模超过 7.5 亿美元,存储容量超过 3.3EB.存储数据的快速增长,使该类存储系统的能耗急剧增加,对该类存储系统进行节能研究是及其必要的.该类存储系统具有特定的数据访问模式和存储特性,例如:以顺序数据访问为主,对随机性能要求不高;对数据的可靠性、存储空间要求较高;以写操作为主,读操作通常回放写操作,称该类存储系统为连续数据存储系统[1,2].

基于连续数据存储系统的存储特性,文献[1]提出了 S-RAID 5 节能磁盘阵列,根据存储应用的实际性能需求,提供合适的局部并行度,而不是像 RAID 5 那样全局并行.局部并行便于在保证性能需求的前提下,调度空闲磁盘,待机实现存储系统节能.在连续数据存储应用中,在满足性能需求及单盘容错条件下,S-RAID 5 可获得显著的节能效果.文献[2]进一步从文件系统、元数据管理等方面对 S-RAID 5 进行了优化,有效提高了 S-RAID 5 的性能和节能效果.

但是,S-RAID 5 的局部并行数据布局会导致 S-RAID 5 基本执行小写(small write)操作,也称读改写(read modify write),即:写新数据时,需要先读取对应的旧数据、旧校验数据,与新数据一起生成新校验数据后再写入新校验数据,严重影响了性能.与 RAID 5 等基本磁盘阵列相比,S-RAID 5 的小写问题更加突出,原因如下[1,2]:

- S-RAID 5 的数据布局及优化策略(如 Cache 策略、文件系统选择与优化),均以如下内容为目标:在充分长的时间内,把 I/O 访问集中在部分并行工作的磁盘上,从而调度其他磁盘待机节能;
- 即使有机会执行重构写,S-RAID 5 通常依然会执行小写,因为执行重构写需要启动所有磁盘,会降低 S-RAID 5 的节能性.

实验测试结果表明,小写使 S-RAID 5 中的单盘有效写带宽的极限值(100%顺序写)不到其最大写带宽的一半.为了提供额定的写性能,S-RAID 5 必须运行更多磁盘以弥补小写带来的性能损失,从而会消耗更多能量.因此,S-RAID 5 的节能效率亟待提高.

已有的小写优化策略主要面向 RAID 5[3,4]、RAID 6[5]等基本磁盘阵列,S-RAID 5 的局部并行数据布局使已有研究难以有效解决其小写问题.舒继武等人[6]提出的 DACO 磁盘架构包括读、写、复合 3 种基本操作,其中,复合操作采用流水技术实现块级数据的读改写操作.容易推论,DACO 磁盘能够解决 S-RAID 5 的小写问题.但该磁盘目前仍处于理论研究阶段,大规模的商业应用前景仍不明朗.

另一方面,连续数据存储应用以顺序访问为主,存在少量随机访问.对存储容量的需求,使磁盘成为首选的存储设备.根据磁盘存储特性可得:即使少量随机访问也会显著降低磁盘性能[7].因此,需要采取合适的措施减少随机访问,充分发挥磁盘的性能.

为此,我们提出一种面向连续数据存储的高效能盘阵——Ripple-RAID,采用了新的局部并行数据布局,并综合运用了以下策略,以实现高性能和高节能效率:

- 地址映射:把非顺序写转化为顺序写;
- 异地更新:把存储空间划分成若干个相等的存储区,其中之一作为影子区,更新存储区 A(源存储区)时,数据实际写入影子区;影子区写满后,修改映射表使它取代存储区 A;下一个循环中,存储区 A 作为影子区,缓存其他存储区的写数据,...,依此类推;
- 渐进生成校验数据:写数据与影子区已有校验数据(初始时无数据)一起生成新校验,随着影子区中数据的增加,校验数据的校验范围也逐渐扩大.生成新校验时无需读取旧数据,当采用流水方式读取影子区已有校验、写入新校验时,可消除读校验数据对性能的影响;
- 分段数据容错:联合影子区、源存储区实现数据恢复,可提供与 RAID 5 相同的单盘容错能力.

在单盘容错的条件下,Ripple-RAID 保持了局部并行的节能性,又解决了局部并行带来的小写问题,具有突出的写性能和节能效率.实验结果表明:在 80%顺序写负载下,当请求长度为 512KB 时,Ripple-RAID 的写性能为

S-RAID 5 的 3.9 倍,是 Hibernator、MAID 写性能的 1.9 倍,PARAID、eRAID 5 写性能的 0.49 倍;而比 S-RAID 5 节能 20%,比 Hibernator、MAID 节能 33%,比 eRAID 5 节能 70%,比 PARAID 节能 72%.连续数据存储中的读操作以数据回放(重复某时间段内的写操作)为主,因此,Ripple-RAID 一般具有与写性能接近的读性能.

1 相关工作

1.1 节能研究现状

大数据(big data)时代正在来临,全世界每天产生 5 万亿字节的数据,这些数据来自各类传感器、视频监控、医疗影像、社交网络等,数据的增长速度远超过摩尔定律的增长速度[8],由此导致存储能耗持续增长.在 Dell PowerEdge 6650 服务器中,存储系统能耗占总能耗的 71%[9].在 EMC Symmetrix 3000 存储系统中,86%的能耗是磁盘驱动器产生的[10].存储系统节能研究是近年来存储领域内的热点问题,并取得一些重要的研究成果[9,11−20].总结已有研究,节能策略可大致分为以下两类.

1.1.1 利用存储设备的空闲时间或负载变动情况节能

现代磁盘一般具有读写、空闲和待机这 3 种工作模式,读写时,盘片全速旋转,进行寻道及数据传输;空闲时,盘片全速旋转但停止了寻道及数据传输;待机时,盘片也完全停止转动.不同模式下的功耗也不同,基本的节能算法是当磁盘空闲时间达到一定值后,把磁盘转入到低功耗的待机模式,当请求到来时再转入到读写模式,称为 TPM(traditional power management)算法[11].

Gurumurthi 等人[11]认为,在企业级工作负载中没有足够长的空闲时间供 TPM 算法利用,因此提出了 DRPM (dynamic rotations per minute)算法:采用动态多转速磁盘,以平均响应时间和请求队列长度为指标,根据工作负载的变动情况动态调整磁盘转速以实现节能.

Carrera 等人[12]通过实验进一步指出:对于性能要求严格的企业级工作负载,采用动态多转速磁盘是唯一可行的节能方法,并提出 LD(load directed)算法:根据工作负载调整磁盘转速,当磁盘工作负载小于低速吞吐量的 80%时,磁盘转入低速模式;大于该值时,转入高速模式.

Zhu 等人[9]基于动态多转速磁盘技术,提出了 Hibernator 节能存储系统:存储系统由多个具有不同转速的 RAID 组成,在存储系统最小能耗和满足性能需求的约束下,利用线性规划方法优化配置了每个 RAID 中的磁盘数量和转速,并在各个不同转速的 RAID 之间动态迁移磁盘,以实现存储系统的最小能耗.

Weddle 等人[13]根据特定工作负载的周期波动特性,借鉴汽车换挡原理,提出了 PARAID 节能磁盘阵列.PARAID 采用倾斜式条带划分方式,在同一组磁盘中构建多级包含不同磁盘数的 RAID 5,根据工作负载的变动情况,动态调度不同 RAID 5 的工作以实现节能.

1.1.2 为存储设备创造空闲时间

“热”数据集中(popular data concentration,简称 PDC)方法[14]根据数据的访问频率进行数据迁移,将访问频率较高的文件迁移到部分磁盘上,而将闲置文件集中到另外一些磁盘上,使闲置文件所在磁盘可长时间待机节能.在视频监控等连续数据存储应用中,数据的访问频率基本服从均匀分布,因此难以应用 PDC 方法实现节能.

MAID(massive arrays of idle disks)[15]使用少量额外磁盘始终运行,作为 Cache 盘保存经常访问的“热”数据,以减少对后端阵列的访问,使后端阵列具有较长的待机时间以实现节能.

在多数据卷存储系统中,Write Off-Loading 方法[16]把待机数据卷(数据卷中的磁盘待机)的写请求暂时重定向到存储系统中某个合适的活动数据卷上,以延长待机数据卷的待机时间,降低磁盘启停的切换频率,并在适当时机恢复重定向的写数据.该方法不适合应用于以写操作为主的连续数据存储中.

Pergamum[17]针对归档存储系统,在每个存储节点添加少量 NVRAM 来存储数据签名、元数据等小规模数据项,从而使元数据请求以及磁盘间的数据验证等操作均可在磁盘待机状态下进行.归档系统的数据访问方式比较简单,如写一次、可能读、新写数据与旧数据不相关[17]等.该方法不适合一般的连续数据存储应用,如在视频监控系统中,当存储空间写满后,会删除最早的视频数据以容纳新数据,需要执行多次写操作.

Li 等人提出的 EERAID[18]将 RAID 内部的冗余信息、I/O 调度策略、Cache 管理策略结合起来,并采用

NVRAM 优化写操作,使冗余磁盘可长时间待机节能.在此基础上,该小组又提出了 eRAID[19],利用 RAID 中的冗余特性来重定向 I/O 请求,进一步延长了冗余磁盘的待机时间,并将系统性能降低,控制在一个可接受的范围内.

毛波等人[20]提出一种绿色磁盘阵列 GRAID,为 RAID10 增加了一个日志盘,周期性地更新镜像磁盘上的数据,将两次更新之间的写数据存放到日志盘和主磁盘上,从而能够关闭所有的镜像磁盘以降低能耗.GRAID 适合对随机性能要求较高的存储应用.

综上,存储系统已有的节能研究主要面向以随机数据访问为主的数据中心,如联机事务处理(on-line transaction processing,简称 OLTP)等,没有充分利用连续数据存储系统的存储特性,因此在连续数据存储中节能效果有限.连续数据存储系统存在着足够的节能优化空间,需要开展细粒度、针对性的节能研究.

1.2 节能研究的发展趋势

存储领域正经历着巨大而深刻的变革,主要包括如下两个方面.

首先,以 NAND Flash 为代表的半导体存储器件开始大规模应用到存储领域,其他类型的存储器,如磁随机存储器(magnetic random access memory,简称 MRAM)、相变存储器(phase change memory,简称 PCM)等也日渐成熟.基于上述存储器的固态盘(solid state disk,简称 SSD)已经成为一种重要的外存储器.

SSD 具有突出的随机读写性能以及低功耗等特点,受限于存储单元集成度和单位存储价格,在可预见的将来,SSD 难以在海量数据存储中彻底取代磁盘[21].磁盘技术仍在不断进步,通过采取各种有效的方法减小寻道时间,磁盘性能保持了 40%的年增长率.磁盘的顺序读、写性能非常突出,如 7 200 转的希捷 ST32000644NS 磁盘,其最大持续数据传输率为 140MB/s,与基于 NAND FLASH 的中端 SSD 相当[22].磁盘在存储容量方面更具突出优势,采用垂直记录技术后,2014 年,记录密度从每平方英寸 1.2 增加到 2.4 万亿比特[22].

其次,大数据日益受到密切关注,大数据的 4 个特点[8],即数据量(volume)大、数据类型(variety)多、价值密度(value)低、处理速度(velocity)快,对云(cloud)中的存储服务——云存储提出了更严格的要求,既要具有海量存储空间,又要提供足够的存储性能.而目前的 SSD、磁盘等外存储器,均无法单独满足以上存储需求.

因此,构建基于 SSD 与磁盘的混合[23]或分层存储系统[24],是节能研究的一个发展趋势.SSD 主要面向随机性、波动性、突发性的工作负载;磁盘存储主要面向稳定的工作负载,或与 SSD 之间进行稳定的数据传输,以顺序数据访问为主,同时满足存储容量需求.可以预测:随着 SSD 更大规模地进入存储领域,云存储中 SSD 层的容量将逐渐扩大,进而成为主要存储层;而后端的磁盘阵列功能将逐渐退化为近似于备份、归档的功能,以顺序访问为主.因此,现在及未来的云存储需要一种面向顺序数据访问的高效能盘阵,在充分发挥磁盘性能的同时,具有更高的节能效率.

只有在满足性能需求的前提下进行节能研究才有意义.根据对平均响应时间、最大响应时间的敏感程度,文献[25]把存储应用分成不同的类型.存储应用性能需求的多样性,要求根据具体应用的存储特性、性能需求开展针对性的节能研究,这是存储系统节能研究的又一个发展趋势.基于存储系统节能研究的以上发展趋势,我们提出了一种面向连续数据存储的高效能盘阵 Ripple-RAID.

2 Ripple-RAID 的实现

Ripple-RAID 的实现主要包括数据布局、写操作方法、数据恢复 3 方面内容,其中,写操作方法又包括地址映射、异地数据更新、基于流水渐进生成校验等内容.

2.1 数据布局

设 Ripple-RAID 由 N 块磁盘组成,每个磁盘平均分成 N+1 个存储区(严格定义应为 kN+1 个存储区,k 为大于 0 的整数,通常取 1,这里以 k=1 为例进行说明),每个存储区称为 Band.N 个相同偏移量的 Band 组成一个 Bank,共组成 N+1 个 Bank,任取其一作为影子 Bank(shadow Bank),其余为基本 Bank.每个基本 Bank 包含 1 个校验 Band、N−1 个数据 Band.在基本 Bank i 中,校验 Band 记为 PBand i,位于磁盘 N−1−i;第 v 个数据 Band 记为 DBand(i,v),当 i+v<N−1 时,DBand(i,v)位于磁盘 v,否则位于磁盘 v+1.其中,0≤i<N,0≤v<N−1.PBand i 的值由公式

(1)通过异或运算求得:

$$\mathrm{PBand}\,i=\bigoplus_{v=0}^{N-2}\mathrm{DBand}(i,v) \tag{1}$$

图 1 给出了一个包含 5 个磁盘的 Ripple-RAID 的总体数据布局.

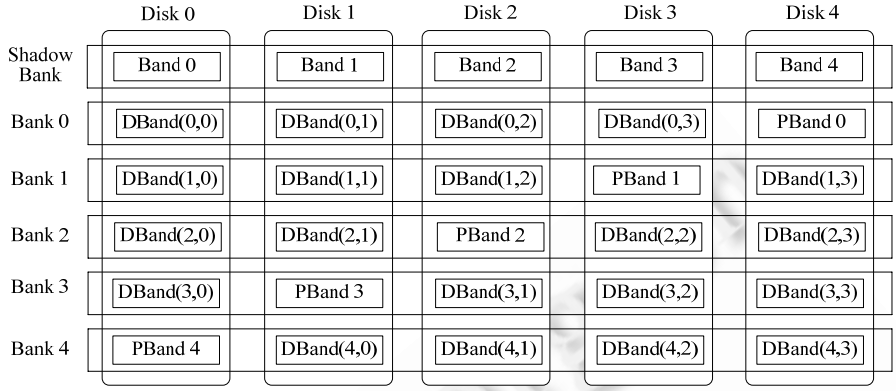

Fig.1 Overview of a 5-disk Ripple-RAID

图 1 5 磁盘 Ripple-RAID 的总体数据布局

令每个 Band 包含 M 个大小相等的 Strip(也称 Chunk,由一些地址连续的数据块组成),每个 Bank 中,相同偏移量的 Strip 组成一个条带(Stripe),这里的 Strip,Stripe 和 RAID 5 中 Strip,Stripe 的含义基本相同[26],但在 Strip 间的地址分配上明显不同.为表述方便,把 PBand 中的 Strip 称为 PStrip.图 2 给出了上述 Ripple-RAID 内基本 Bank 0 的数据组织方式.

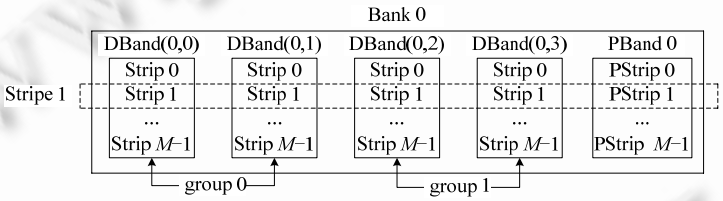

Fig.2 Data organization of the Bank 0 in the 5-disk Ripple-RAID

图 2 5 磁盘 Ripple-RAID 内 Bank 0 的数据组织方式

为了提供合适的性能,Ripple-RAID 采用如下局部并行数据来布局:把每个基本 Bank 中的 N–1 个数据 Band 平均分成 P 组,每组包含 Q 个.每组中偏移量相同的 Strip 能够被并行访问,每个 Stripe 中仅部分 Strip 提供并行性.如图 2 所示,Bank 0 包含 2 个组(group),每组含有 2 个数据 Band(P=2,Q=2),其中,group 0 包含 DBand(0,0)和 DBand(0,1),group 1 包含 DBand(0,2)和 DBand(0,3).在 group 0 中,DBand(0,0)的 Strip 1 和 DBand(0,1)的 Strip 1 并行工作,不像 RAID 5 那样,Stripe 中所有 Strip 并行工作.

Ripple-RAID 仅对基本 Bank 进行分组,影子 Bank 不参与分组,也不参与编址,对 Ripple-RAID 的上层应用是透明的.在组地址分配上,Ripple-RAID 采用了适度的贪婪策略:在每个基本 Bank 中,序号相邻的组的逻辑地址相邻.如图 3 所示,在 Bank 0 中,group 0 与 group 1 的逻辑地址相邻.设 $NumBlk_{\mathrm{Strip}}$ 为 Strip 包含的数据块数,则 Bank i,group p,Band q 中第 m 个 Strip 的逻辑地址见公式(2):

$$\mathrm{Strip}_{i,p,q,m}(addr)=NumBlk_{\mathrm{Strip}}(M\cdot Q\cdot P\cdot i+M\cdot Q\cdot p+Q\cdot m+q) \tag{2}$$

这里,$0\leqslant p<P$,$0\leqslant i<N$,$0\leqslant q<Q$,$0\leqslant m<M$.

Ripple-RAID 的数据布局和编址方式能够提供足够的并行度,并且对于连续数据存储应用,可保证 I/O 请求在很长的时间内集中在一个或几个 group 中(1 个 group 可包括几个 DBand,而 Ripple-RAID 中的 DBand 又足够

大),其他多数磁盘有足够长的待机时间,可调度到待机模式以节约能耗.

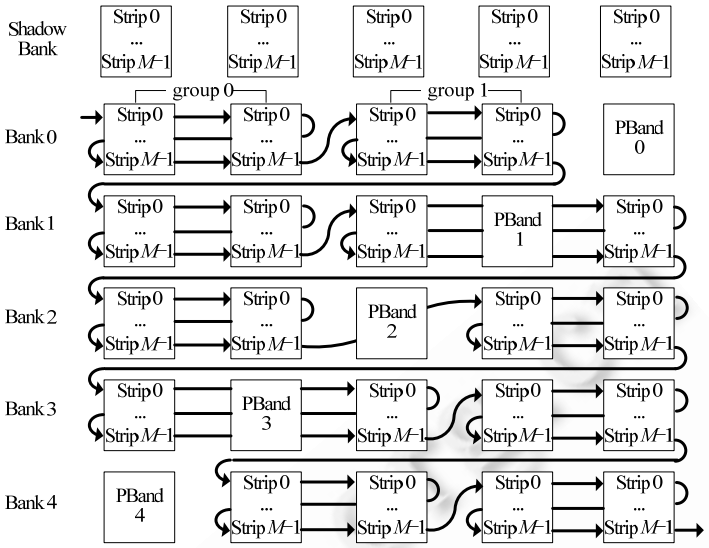

Fig.3 Address allocation strategy of the 5-disk Ripple-RAID

图 3 5 磁盘 Ripple-RAID 的地址分配策略

需要阐明如下问题:在连续数据存储应用中,通过调整 RAID 5 中 Strip 的大小,不能获得与 Ripple-RAID 类似的节能效果.原因如下:

- 如果 RAID 5 的 Strip 足够大,执行读写请求时,单个磁盘的访问时间将被延长,在很长一段时间内仅有一个磁盘被访问,因而难以提供足够的并行性以满足性能需求;
- 如果 RAID 5 的 Strip 不足够大,则同一条带中的 Strip 在较短时间内会被频繁访问,没有机会调度磁盘待机节能,

在 Ripple-RAID 中,Strip 大小没有限制,可以根据实际需要进行设置.

2.2 写操作方法

Ripple-RAID 的写操作方法综合运用了地址映射、异地数据更新、渐进生成校验等策略,生成校验数据时无需读取旧数据,当采用流水方式读取已有校验(与旧校验不同)、写入新校验时,可有效解决局部并行带来的小写问题.此外,地址映射把非顺序写转换成顺序写,又进一步提升了 Ripple-RAID 的写性能.

2.2.1 地址映射

顺序数据访问能够充分发挥磁盘性能,如:日志文件系统(log-structured file system)[27]创建或改写文件时,新数据以顺序写方式添加到日志中;网络文件系统 Zebra[28]把每个客户端的写数据组成一个连续的日志,条带化后,分布存储到各服务器上.以上方法均把多个小的、随机写操作转换为大的、顺序写操作,以提高存储系统的写性能.

连续数据存储系统非常适合进行地址映射.首先,该类系统以写操作为主,把非顺序写转换为顺序写后,可显著提高写性能和整体性能;其次,读操作以数据回放为主,即,重复以前某时间段内的写操作,如视频监控中的视频回放等,通常可获得与写性能接近的读性能.

常见的地址映射方法有单块映射和块组映射.单块映射需要记录每个数据块的映射关系,多块映射时效率不高,典型应用有 NILFS[29]文件系统;块组映射以若干个连续数据块为单位进行映射,多块映射时效率高,但存在块组数据的“读写”问题,即,改写块组中部分数据时,需要读取其余未修改数据,与新数据一起重新进行地址映射,典型应用有 HP AutoRAID[30],块组大小为 64KB.

连续数据存储系统以写新数据为主,较少进行改写操作,适合采用块组映射.地址映射信息为存储容量的8/(1024×x),其中,8 个字节(64 位)记录一个块组地址,x 为块组大小以 KB 为单位.当 Ripple-RAID 的存储容量为30TB、块组大小为 64KB 时,地址映射信息仅为 3.67GB,适合采用 SSD 进行存储,运行时甚至可以完全调入内存,以加快读、写操作中的地址转换速度.

把非顺序写转换为顺序写,需要面对垃圾回收(garbage collection)[27]问题,垃圾存储空间是由改写操作产生的,在连续数据存储中,如视频监控、CDP、备份、归档等应用,改写的数据量不大,可在负载较轻时进行垃圾回收;如果追求性能,也可牺牲少量存储空间而忽略垃圾回收.

2.2.2 异地数据更新

地址映射把非连续的虚拟地址映射为连续的物理地址,并在映射表中记录映射关系.其中,虚拟地址为应用程序发来的读写请求地址,物理地址为数据在 Ripple-RAID 内的存储地址(影子 Band 不参与编址).在此基础上,Ripple-RAID 执行异地数据更新:向某物理地址写数据时,数据不直接写入该地址,而是写入其影子地址(影子 Bank 中与其偏移量相同的地址),并在适当时候修改映射表,令影子地址取代该物理地址.异地数据更新在 NAND Flash 中也有应用,因 NAND Flash 无法实现数据的就地改写[22].

假设 Ripple-RAID 由 N 块磁盘组成,划分出 N+1 个 Bank,任取其中之一作为影子 Bank,其余为基本 Bank,则 Ripple-RAID 的异地数据更新过程如下:

(1) 向某基本 Bank(称源 Bank)写数据时,数据并不直接写入该 Bank,而是写入影子 Bank;

(2) 根据写入数据、本次循环中影子 Bank 中已写数据的校验数据,生成影子 Bank 的新校验数据;

(3) 如果影子 Bank 未写满,转到步骤(1);

(4) 否则,修改地址映射关系,令影子 Bank 取代源 Bank,本次循环结束;

(5) 被取代的源 Bank 此时无映射关系,可在下一循环中作为影子 Bank.

在以上写操作过程中,由于进行了地址映射,所以是依次向每个基本 Bank 顺序写入数据的,不会同时向两个基本 Bank 写数据,也不会在一个基本 Bank 未写满的情况下,向另外一个基本 Bank 写数据.

2.2.3 渐进式生成校验

影子 Bank 的校验数据是根据本次循环中已写数据生成的,称为局部校验数据(不同于旧校验).写新数据时,可根据新数据、局部校验数据计算新校验数据,无需读取旧数据.随着写数据的增加,局部校验数据的校验范围也渐进扩大,直至扩展到整个影子 Bank.局部校验数据的校验范围以及新数据的写入,如水中涟漪一样向前推进,因此,该盘阵命名为 Ripple-RAID.

I. 相关流水(relevant pipeline)方式

渐进生成校验数据时无需读取旧数据,仅需读取局部校验数据,因此可增加一个辅助存储设备,与影子 Bank 中校验数据所在磁盘,以流水方式生成新校验(1 个读局部校验数据,1 个写新校验),此时,可有效消除读校验数据对写性能的影响.

图 4 给出了一个 Ripple-RAID 的写操作示例,其中,每个 Bank 包含 3 个 group,IParity (intermediate parity)为辅助存储设备,暂存影子 Bank 中的局部校验数据.与 PBand 容量相同,阴影部分为局部校验数据的校验范围,具体执行过程如下:

(1) 向任一基本 Bank(称源 Bank)的 group 0 写数据时,数据实际写入影子 Bank 的 group 0,并生成 group 0 的校验,写入影子 Bank 的 PBand,如图 4(a)所示;

(2) group 0 写满后,向源 Bank 的 group 1 写数据时,数据实际写入影子 Bank 的 group 1,并根据写数据、局部校验(group 0 的校验,在影子 Bank 的 PBand),生成新校验(group 0,group 1 的校验),写入 IParity,如图 4(b)所示;

(3) group 1 写满后,向源 Bank 的 group 2 写数据时,数据实际写入影子 Bank 的 group 2,并根据写数据、局部校验(group 0,group 1 的校验,在 IParity),生成新校验(group 0,group 1,group 2 的校验),写入影子 Bank 的 PBand,如图 4(c)所示;

(4) 影子 Bank 写满后,修改映射表,令其取代源 Bank,而源 Bank 作为下一循环中的影子 Bank.

为保证最后生成的校验数据写入影子 Bank 的 PBand,需按如下规则流水:若影子 Bank 的 group 数为奇数,则首先向 PBand 写校验数据,如图 4(a)所示;否则,首先向 IParity 写校验数据.当 IParity 采用低功耗的 SSD 时,其能耗增加可以忽略.影子 Bank 中校验数据所在磁盘,与辅助存储设备一起进行流水,所以该流水方式称为相关流水.

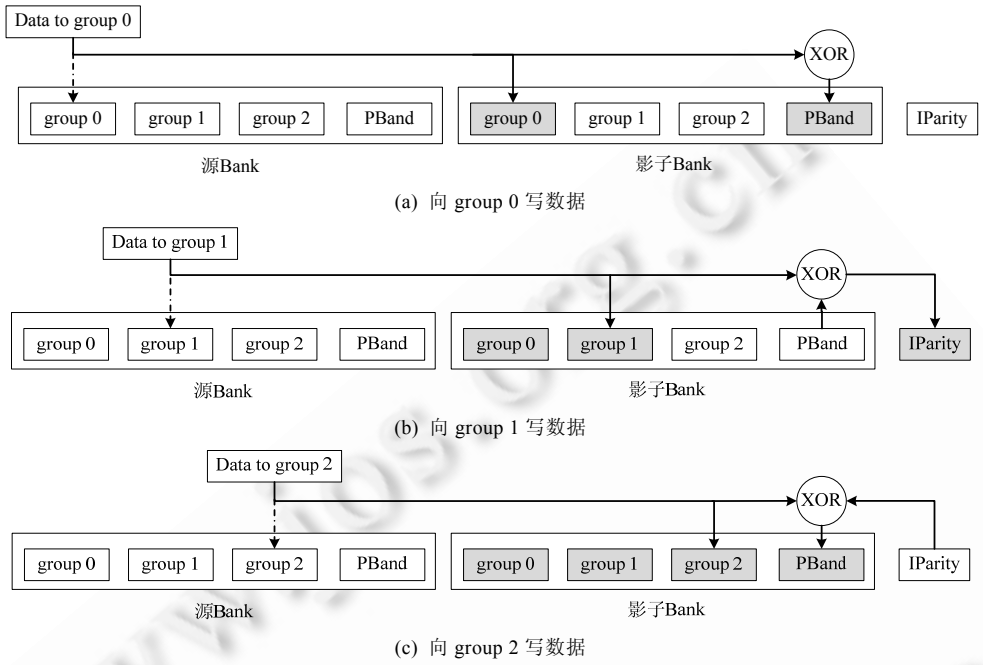

Fig.4 Write of Ripple-RAID in relevant pipeline for parity

图 4 相关流水的 Ripple-RAID 写操作

II. 基于 SSD 的非流水方式

采用 SSD 作为 IParity 时,可不采用流水方式生成校验数据:从 IParity 读局部校验数据,新校验数据也写入 IParity,直至写最后 group 数据时,从 IParity 读局部校验数据,并将新校验数据写入磁盘.影子 Bank 中校验数据所在磁盘大部分时间也可待机,节能效果将进一步得以提升,但生成校验时需要同时读、写 IParity,对写性能有一定影响,称该方式为基于 SSD 的非流水方式.

III. 无关流水(irrelevant pipeline)方式

为使影子 Bank 中校验数据所在磁盘大部分时间也可待机,进一步提高节能效率的同时又不影响性能,可采用如下流水方式:设置两个辅助存储设备 IParity 1 和 IParity 2,轮流从其中之一读局部校验数据,向另一个写新校验数据,直至生成影子 Bank 的最终校验数据,再将其写入磁盘.影子 Bank 中校验数据所在磁盘不参与流水,因此该流水方式称为无关流水.当 IParity 1 和 IParity 2 均采用低功耗的 SSD 时,其能耗增加可以忽略.

图 5 给出了一个基于无关流水的 Ripple-RAID 写操作示例,其中每个 Bank 包含 3 个 group,IParity 1 和 IParity 2 为辅助存储设备,用于流水生成影子 Bank 的局部校验数据,容量与 PBand 相同,阴影部分为局部校验数据的校验范围.该写过程与相关流水方式相似,仅有几点不同之处:

(1) 向 group 0 写数据时,生成的局部校验(group 0 的校验)写入 IParity 1,如图 5(a)所示;

(2) 向 group 1 写数据时,根据写数据、局部校验(group 0 的校验,在 IParity 1),生成新校验(group 0,group 1 的校验),写入 IParity 2,如图 5(b)所示;

(3) 向 group 2 写数据时,根据写数据、局部校验(group 0,group 1 的校验,在 IParity 2),生成最终校验(group

0,group 1,group 2 的校验),并把最终校验写入影子 Bank 的 PBand,如图 5(c)所示.

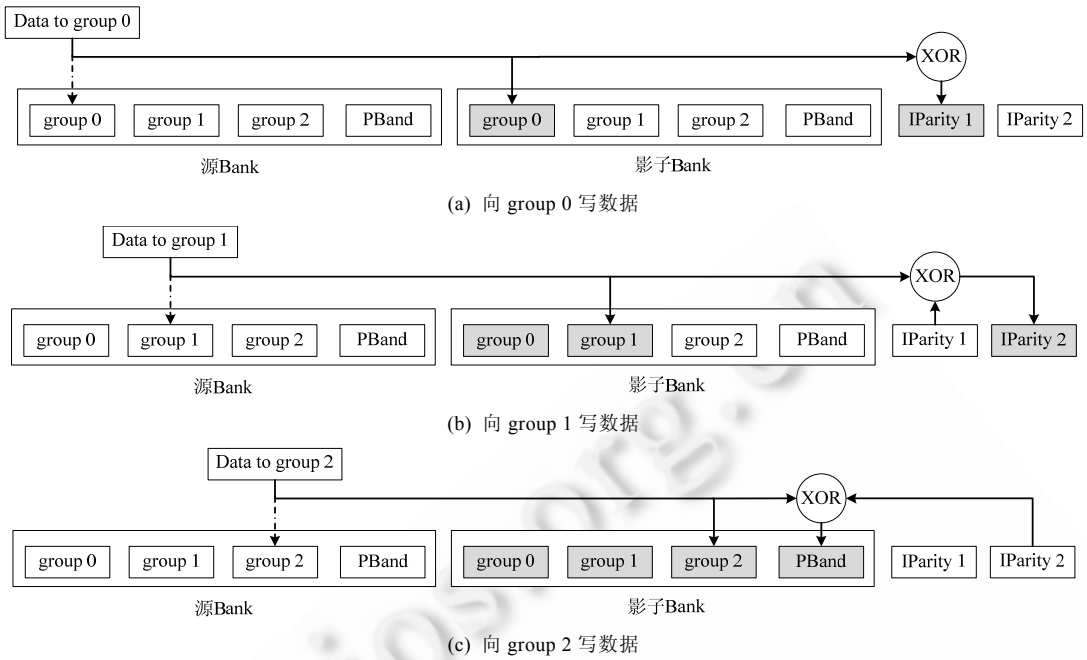

Fig.5 Write of Ripple-RAID in irrelevant pipeline for parity

图 5 无关流水的 Ripple-RAID 写操作

2.3 数据容错

性质 1. Ripple-RAID 具有单盘容错能力.

证明:假设 Ripple-RAID 包含 N 块磁盘,其中的 N+1 个 Bank 分为 1 个影子 Bank 和 N 个基本 Bank,每个基本 Bank 包含 N–1 个数据 Band(分成 P 组,每组 Q 个)和 1 个校验 Band,Band 大小均为 M 个 Strip.按当前状态(是否正在被更新)把基本 Bank 分为活跃 Bank(active Bank)和睡眠 Bank(inactive Bank)两类,影子 Bank 的数据组织方式与活跃 Bank 相同.

由于地址映射后执行顺序写,因此在确定时间内,只有 1 个基本 Bank 被更新,即:只有 1 个活跃 Bank,其余皆为睡眠 Bank.为了便于理解,证明过程中给出了 1 个 Ripple-RAID 示例,包括 7 块磁盘,分为 1 个影子 Bank 和 7 个基本 Bank,每个基本 Bank 中的数据 Band 分为 3 组,每组包含 2 个数据 Band,即 P=3,Q=2.

情况 1:对于睡眠 Bank,其任一条带 Stripe m 包含 $P\cdot Q$ 个 Strip m,被平均分成 P 组,每组包含 Q 个,$0\leqslant m<M$,如图 6 所示,假设当前睡眠 Bank 为 Bank 1.

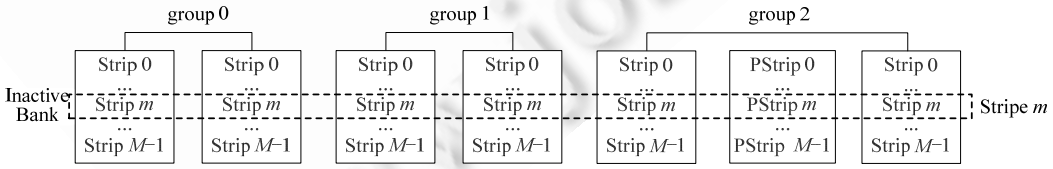

Fig.6 Data recovery of inactive Bank in Ripple-RAID

图 6 Ripple-RAID 中睡眠 Bank 的数据恢复

根据 Ripple-RAID 的写过程,可得公式(3)成立,当根据公式(3)生成 PStrip m 后,直到该睡眠 Bank 成为活跃 Bank 之前,该条带中的 $P\cdot Q$ 个 Strip m 以及 PStrip m 均未被修改过,已建立的校验关系有效.因此,任一磁盘出现

故障时,可根据公式(3)实现数据恢复.

$$\mathrm{PStrip}\,m=\underbrace{(\overbrace{\mathrm{Strip}\,m\oplus\ldots\oplus\mathrm{Strip}\,m}^{Q\text{个}})\oplus(\overbrace{\mathrm{Strip}\,m\oplus\ldots\oplus\mathrm{Strip}\,m}^{Q\text{个}})\oplus\ldots\oplus(\overbrace{\mathrm{Strip}\,m\oplus\ldots\oplus\mathrm{Strip}\,m}^{Q\text{个}})}_{P\text{组}} \tag{3}$$

情况 2:对于活跃 Bank,以最后一次局部并行写为分界线,分界线之前为已写区,其后为待写区.设分界线位于第 p 组中偏移量为 m 的 Strip 之后,$0\leqslant p<P$,$0\leqslant m<M$.例如,图 7 中活跃 Bank(假设为 Bank 0)的分界线位于第 1 组(p=1)中偏移量为 m 的 Strip 之后(活跃 Bank 的已写区为正体,待写区为正体加粗).

I. 对于活跃 Bank 的已写区数据,由于对应的新数据及其校验数据全部写入影子 Bank(斜体),所以在影子 Bank 中具有完整、有效的校验关系.对于影子 Bank 中的条带 Stripe k,当 $0\leqslant k\leqslant m$ 时,其校验关系见公式(4).

$$\mathrm{PStrip}\,k=\underbrace{(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\oplus\ldots\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})}_{p+1\text{组}} \tag{4}$$

如图 7 中影子 Bank 的 Stripe 0 所示,有 2 组(p+1=2)共 4 个(每组 2 个)Strip 0 参与校验,×表示该数据未参与本条带校验运算.当 $m<k<M$ 时,在影子 Bank 中存在条带 Stripe k(仅当 $p\geqslant 1$ 时存在该情况),其校验关系见公式(5).

$$\mathrm{PStrip}\,k=(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\oplus\ldots\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}}) \tag{5}$$

如图 7 中影子 Bank 的 Stripe M–1 所示,有 1 组(p=1)共 2 个(每组 2 个)Strip M–1 参与校验.因此,当任一磁盘出现故障时,对于活跃 Bank 的已写区数据,可根据影子 Bank 中条带的位置,利用公式(4)或公式(5)实现数据的恢复.

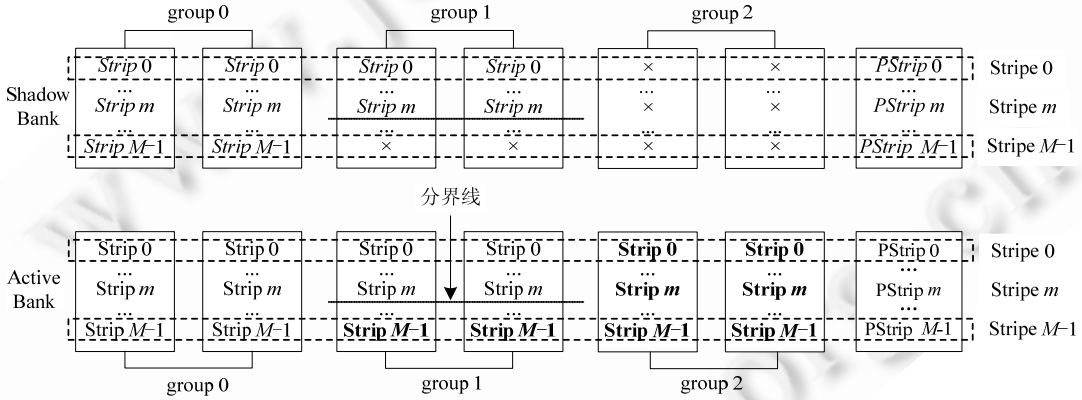

Fig.7 Data recovery of active Bank in Ripple-RAID

图 7 Ripple-RAID 中活跃 Bank 的数据恢复

II. 对于活跃 Bank 的待写区数据,当 $0\leqslant k\leqslant m$ 时,条带 Stripe k 的校验关系见公式(6).

$$\mathrm{PStrip}\,k=\underbrace{(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\oplus\ldots\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})}_{(p+1)\text{组}}\oplus\ldots\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\Big)_{P\text{组}} \tag{6}$$

其中包括 p+1 组位于活跃 Bank 已写区的数据,如图 7 中活跃 Bank 的 Stripe 0 所示,有 2 组(p=1)数据位于已写区.当 $m<k<M$ 时,条带 Stripe k 的校验关系见公式(7).

$$\mathrm{PStrip}\,k=\underbrace{(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\oplus\ldots\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})}_{p\text{组}}\oplus\ldots\oplus(\overbrace{\mathrm{Strip}\,k\oplus\ldots\oplus\mathrm{Strip}\,k}^{Q\text{个}})\Big)_{P\text{组}} \tag{7}$$

其中包括 p 组位于活跃 Bank 已写区的数据,如图 7 中活跃 Bank 的 Stripe M–1 所示,有 1 组(p=1)数据位于已写区.由于活跃 Bank 中已写区数据在校验关系建立后并没有被真正修改过(新数据被写入影子 Bank 的对应位置),公式(6)、公式(7)表示的校验关系仍然有效,因此,当任一磁盘出现故障时,对于活跃 Bank 的待写区数据,可根据活跃 Bank 中条带的位置,利用公式(6)或公式(7)实现数据的恢复.

综合情况 1、情况 2 可得,Ripple-RAID 具有单盘容错能力.Ripple-RAID 的分界线(最后一次局部并行写位置)对于实现数据恢复至关重要,因此需要记录到元数据中,以保证数据恢复的正确执行.上述容错能力的证明过程也是 Ripple-RAID 的数据恢复过程,可得其数据恢复时间与 RAID 5 和 S-RAID 5 相当. □

3 实验测试

3.1 实验环境

为了测试 Ripple-RAID 的性能和节能效果,利用 Linux 2.6.26 内核中的 MD(multiple device driver)模块构建了一个 Ripple-RAID 的原型系统.监控进程 Diskpm 对磁盘进行节能调度,采用 TPM 调度算法,当磁盘空闲时间达到 120s 时,调度该磁盘待机.采用文献[2]中的 Cache 策略,以减少少量读操作对待机磁盘的访问.

基于典型的连续数据存储应用——视频监控进行了性能、节能测试.模拟了一个 32 路视频监控系统,采用 D1 视频标准(平均码率为 2Mb/s),需要保存 24h/天×30 天的视频数据,数据量为 20.74TB.每隔指定时间(实验中为 10 分钟)在存储设备上创建 32 个视频文件,分别保存该时间内的各路视频数据,视频数据以添加(append)方式写入视频文件,当存储空间不够时,删除最早存储的视频数据.

选取了几种典型的 RAID 节能方法与 Ripple-RAID 进行冗余磁盘、性能和节能比较,具体包括 Hibernator, PARAID,eRAID 5,MAID 以及 S-RAID 5.功耗测量系统如图 8 所示,包括 1 台运行 Linux 2.6.26 的存储服务器、磁盘阵列(类型及盘数需分别设定)、测控计算机、电流表以及电源等部分.存储服务器配置如下:Intel (R) Core (TM) i3-2100 CPU,8GB 内存,主板型号为 ASUS P8B-C/SAS/4L,主板上集成的 LSI 2008 SAS 存储控制器在背板上扩展出 32 个 SAS/SATA 盘位,选用 2TB 的希捷 ST32000644NS 磁盘.

磁盘功率测量如图 9 所示,利用电流表测量磁盘工作电流,测控计算机负责设定电流采用频率,并在测量结束后读取测量值,电流表通过 LAN 线与测控计算机相连.采用 GW PPE-3323 高精度稳压电源为磁盘提供+5V 和+12V 电压,利用 Agilent 34410A 数字万用表分别测量并存储其电流值.最后,测控计算机根据电流、电压值计算出功率值及总功耗.

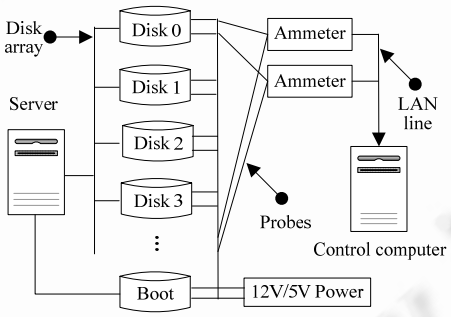

Fig.8 Power measurement framework

图 8 功耗测量系统

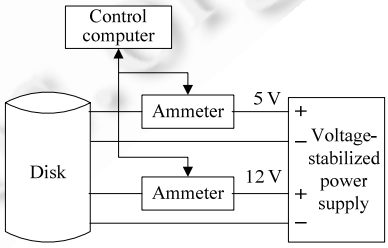

Fig.9 Power measurement of a disk

图 9 磁盘功率测量

每种节能方法的功耗测量时间为 24 小时,电流采样频率为 5Hz,Strip(也称 Chunk)大小为 64 KB.与 S-RAID 5 不同,由于在块层进行了地址映射,Ripple-RAID 的节能效果对文件系统的选择不敏感.但为了公平比较,选择了相同的 NILFS 文件系统,该文件系统非常适合视频监控等连续数据存储应用.

3.2 冗余磁盘

需要保存的视频数据量为 20.74TB,所以需要 11 块容量为 2TB 的磁盘.考虑到文件系统对存储空间的额外消耗,约为存储空间的 10%,取 12 块磁盘保存基本数据.

Ripple-RAID 需要 1 块磁盘的校验数据,共需 13 块磁盘.当映射块组大小为 64KB 时,24TB (2TB×12)的存储空间需要 2.93GB 的地址映射信息(计算方法见第 2.2.1 节),采用镜像保护方式并取整为 6GB;辅助存储设备 IParity 的容量与 PBand 相同,大小为 140GB(2000GB/(13+1));影子 Bank 需求(N–1)/(N+1)块磁盘的存储空间,上述配置能够满足.

Hibernator 把不同转速的磁盘组成不同的 RAID,由于磁盘有运行和待机两种转速,需要构建 2 个 RAID,分别处于运行和待机状态,并根据性能需求在 2 个 RAID 间迁移数据盘.因此,需要 2 个磁盘的校验信息,共需 14 块磁盘;PARAID 中跨越磁盘数最少的逻辑 RAID 的节能效果最好,由于每级逻辑 RAID 都需要保存 1 份完整的存储数据,因此在最节能逻辑 RAID 中要保存 12 块磁盘的数据量,加上 1 块磁盘的校验信息,共需 13 块磁盘;eRAID 5 需要 1 块盘的校验信息,共需 13 块磁盘;MAID 由 2 个磁盘阵列组成,前端阵列保存“热”数据以减小对后端阵列的访问,全部数据保存在后端阵列中,前端阵列为由 4 块磁盘组成的 RAID 5(原因见第 3.3 节),后端阵列为由 13 块磁盘组成的 RAID 5,共需 17 块磁盘;S-RAID 5 需要 1 块磁盘的空间存储校验信息,共需 13 块磁盘.

综上,配置以上节能 RAID 所需的冗余磁盘数,除 MAID 略高(5 块)外,其余均为 1 或 2 块,Ripple-RAID 接近但小于 2 块.此外,Ripple-RAID 以相关流水生成校验时需要 146GB(IParity 及映射信息)的 SSD,无关流水时需要 286GB(IParity 1,IParity 2 及映射信息)的 SSD,不超过总存储容量的 2%,在海量数据存储中是可以接受的.

3.3 性能测试

32 路 D1 标准的视频监控系统所需基本写带宽为 8MB/s(32×2Mb/s),写性能要求不高.但是为了保证具有足够的性能裕量,要求每种节能方法至少提供 3 块磁盘(不包括校验数据所在磁盘)的并行度.对于 Ripple-RAID 与 S-RAID 5,每个 Bank 中的 12 个 DBand 被分为 4 组,每组 3 个并行工作(P=4,Q=3);Hibernator 需要把 3 块数据磁盘迁移到运行阵列,与校验盘组成 RAID 5;对于 MAID,取其前端阵列为 4 块磁盘(校验数据占 1 块磁盘的空间)组成的 RAID 5.

Ripple-RAID 分别以相关流水(1 块 SSD)、无关流水(2 块 SSD)方式生成校验数据,采用的 SSD 型号为 PX-160M3,容量为 160GB.选取每种阵列的基本工作状态,也是最佳节能状态来测试性能.Ripple-RAID 与 S-RAID 5 的测试逻辑地址范围指定在 1 个分组之内;Hibernator 的测试对象为其运行阵列;MAID 的测试对象为其前端阵列;PARAID 的测试对象为其最节能的那一级 RAID 5.

首先,利用 Iometer 测试写性能,测得各节能阵列在 80%顺序写、随机写负载下的写性能,分别如图 10(a)、图 10(b)所示,Ripple-RAID 具有突出的写性能,相关流水、无关流水时的写性能基本相同,统一记作 Ripple-RAID.在 80%顺序写负载下,当请求长度为 512KB 时,与并行盘数相同的节能阵列相比,Ripple-RAID 的写性能分别为 S-RAID 5 的 3.9 倍,是 Hibernator 和 MAID 写性能的 1.9 倍;与 12 磁盘并行的 PARAID 和 eRAID 5 相比,Ripple-RAID 的写性能达到了前两者的 49%.

随机负载增加时,Ripple-RAID 写性能会更加突出.在随机写负载下,其写性能远高于并行盘数相同的 S-RAID 5、Hibernator 以及 MAID,而与 12 磁盘并行的 PARAID 和 eRAID 5 的写性能相当.Ripple-RAID 突出的写性能得益于有效消除了局部并行带来的小写问题以及通过地址映射把非顺序写转换成了顺序写.S-RAID 5 的写性能最低,主要是由于其局部并行数据布局带来的小写问题,严重影响了写性能.

Ripple-RAID 读性能取决于地址映射后的数据分布情况,顺序读可被映射为随机读(概率大),随机读也可被映射为顺序读(概率小),因此难以给出准确的读性能对比测试.但可以给出 Ripple-RAID 中地址转换延迟对读性能的影响,我们做了一个地址平移变换,把所有读请求平行映射到另外一个读区间.

在此基础上,利用 Iometer 测试了读性能,各节能阵列在 80%顺序读、随机读时的读性能分别如图 10(c)、图 10(d)所示,Ripple-RAID 的读性能略低于并行盘数相同的 S-RAID 5、Hibernator 和 MAID,是由于 Ripple-RAID

的地址转换引起一定的时间延迟;读性能远低于 PARAID 和 eRAID 5,是由于 Ripple-RAID 提供了 3 磁盘的并行度,而后者均提供了 12 磁盘的并行度.

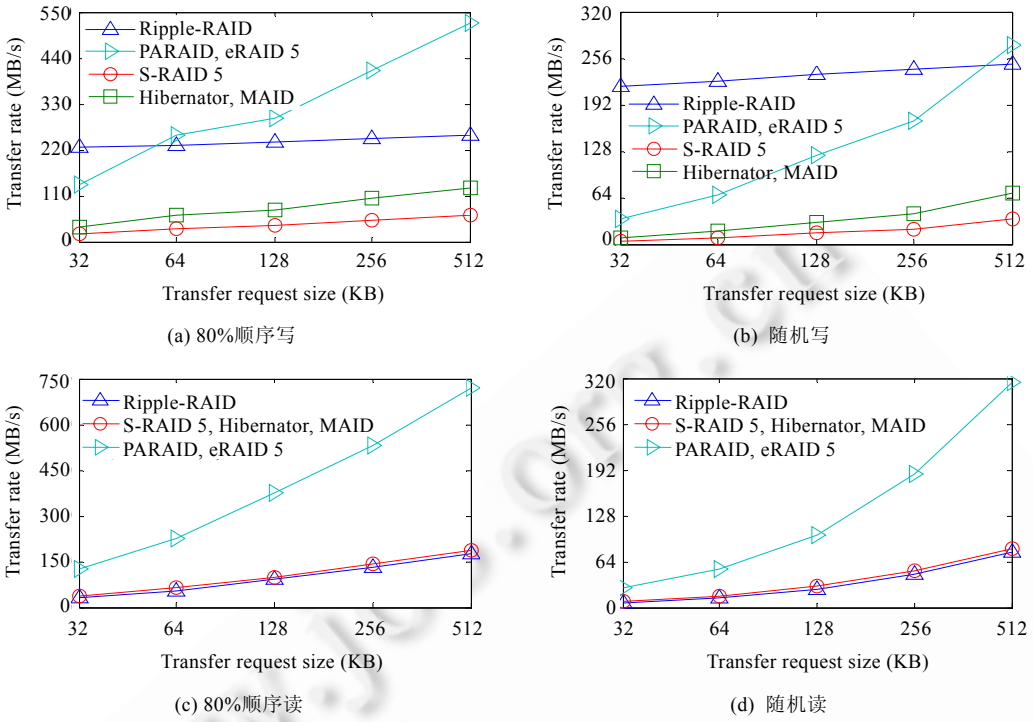

Fig.10 Transfer rate of different energy-saving approaches

图 10 不同节能方法的数据传输率

以上读性能测试结果,仅具有一定的参考价值.在实际的连续数据存储应用中,由于读操作以数据回放为主,如视频回放、利用 CDP 进行系统还原、读取归档数据等,一般会执行顺序读(重复某段时间内的写操作),此时,Ripple-RAID 的读性能将与写性能接近,远高于以上测试结果.

Ripple-RAID 采用流水技术渐进生成校验数据(如图 4、图 5 所示),其中,新数据写入磁盘,导致 Ripple-RAID 的性能主要取决于磁盘性能.缓存新校验与写新数据并行.因此,把缓存新校验的 SSD 改成磁盘,不会显著影响其性能,采用 SSD 主要为了提高节能效率.

Ripple-RAID 把磁盘存储区分成若干组,组内局部并行,组间可独立工作.对于读写混合型负载,容易解耦成独立的读写操作,具体如下:如果读写操作分别位于 Ripple-RAID 中可并行的组,则调度对应的组运行,此时,Ripple-RAID 的总性能基本等于各组读、写性能之和;否则,先执行读操作,同时缓存写数据到相关设备(如低功耗的 SSD),并在读操作结束后回迁写数据,此时,Ripple-RAID 的性能等于该组的读、写性能.

综上,Ripple-RAID 中单个分组的读、写性能能够反映 Ripple-RAID 的基本性能.为了进一步验证该分组方式能否满足性能需求,我们进行了实际数据读写测试.向 Ripple-RAID 写入视频数据,然后检验写入数据的正确性,同时进行视频回放.测试结果表明,该 Ripple-RAID 能够正确写入 32 路 D1 标准的视频数据以及正确回放记录的数据.为了避免直接从内存缓冲区读取回放数据,回放的是 1 小时以前的监控数据.

3.4 节能测试

对上述 Ripple-RAID、S-RAID 5、Hibernator、PARAID、MAID 以及 eRAID 5 分别进行 24 小时节能测试,测试结果如图 11 所示.Ripple-RAID 的节能效果最好,采用无关流水生成校验数据时,24 小时平均功耗约为

2.4×10^6J,比 S-RAID 5 节能 20%,比 Hibernator 和 MAID 节能 33%,比 eRAID 5 节能 70%,比 PARAID 节能 72%;采用相关流水生成校验数据时,可节省 1 块 SSD(160GB),性能与无关流水时基本相同,但功耗有所增加.此时,Ripple-RAID 的功耗与 S-RAID 5 基本相同.

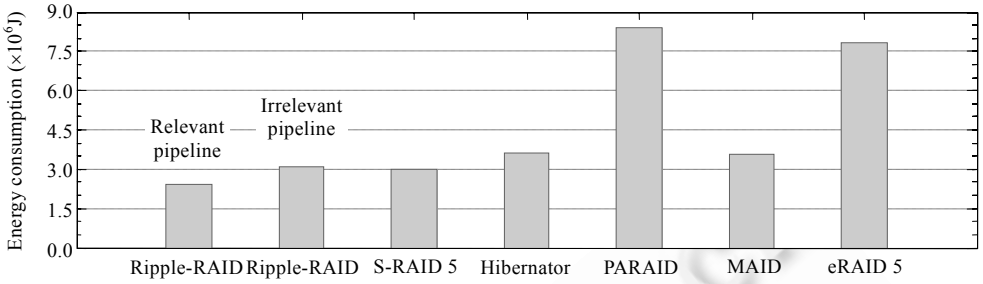

Fig.11 24-h energy consumption of different energy-saving approaches

图 11 不同节能方法的 24 小时功耗

S-RAID 5 的节能效果与 Ripple-RAID 接近,但写性能远低于 Ripple-RAID,因此,其性能裕量远小于 Ripple-RAID.与 Hibernator 相比,Ripple-RAID 没有把存储空间划分为多个不同转速的子阵列,省略了子阵列间的磁盘迁移、数据重构过程.PARAID 的能耗最高,表明该节能阵列不适合数据密集型存储应用,而适用于具有较多空闲存储空间的存储系统.与 MAID 相比,Ripple-RAID 把存储数据直接写入局部并行的存储阵列,省略了前端数据缓冲过程及相关存储设备.eRAID 5 主要面向随机数据访问,最多仅能节省 1 块磁盘的功耗(关闭校验数据所在磁盘).

4 磁盘状态转换的影响

本节将从写、读两方面分析磁盘状态转换对性能、节能和存储设备寿命的影响.Ripple-RAID 执行顺序写,可以预启动将要访问的待机磁盘来消除状态转换对写性能的影响.下面将分析写操作引起的状态转换对节能、存储设备寿命的影响.实验中的 Ripple-RAID 采用 P=4,Q=3 的分组方式,Band 大小为 140GB,group 大小为 420GB (140GB×Q).32 路 D1 视频监控系统每小时产生 28.8GB 的数据,所以 group 的切换周期为 420/28.8=14.6 小时;磁盘状态转换周期(运行→待机→运行)为 58 小时(14.6×P).采用流水方式生成校验数据时,SSD 的擦写周期为 29 小时(group 切换周期×2),若 SSD 可擦写 10 万次,则其寿命为 333 年.保持以上配置不变,写带宽增加到 200MB/s(原来的 25 倍),磁盘状态转换周期为 2.3 小时,SSD 寿命为 13 年.

继续增大写带宽,单个 Ripple-RAID 将难以满足存储容量需求(S-RAID 5,Hibernator 情况相同).以 200MB/s 为例,24 小时的数据量为 17TB,以 12 块 2TB 的磁盘构建 RAID 时(RAID 中磁盘数一般为 10 块左右,若过多,则难以保证安全性),去除文件系统的消耗,有效存储空间约为 21TB,仅能存储 30 小时的数据.连续数据存储系统一般需要较长时间保存数据,如我国规定:一般场所的视频监控数据保存 15 天,重要场所的监控数据保存 30 天以上.此时,需要采用多个 RAID 提供存储容量,分为两种情况:(1) 多个 RAID 并行,写带宽分配到各个 RAID,单个 RAID 的写带宽将有所下降;(2) 多个 RAID 串行(交替工作),每个 RAID 中的磁盘状态转换周期依然会足够大.因此,可忽略写操作引起的状态转换对节能、存储设备寿命的影响.

连续数据存储系统以写操作为主,包含少量读操作,可分为如下两类:(1) 读取文件系统元数据、RAID 配置信息等,数据量较小,读数据分布具有一定规律;(2) 数据回放时的读操作,数据量一般较大,读数据随机分布.采用文献[2]提出的优化策略,可有效消除第(1)类读操作引起的磁盘状态转换问题.第(2)类读操作的执行频率通常很低,并以顺序读为主(回放写操作),所引起的磁盘状态转换对性能、节能以及磁盘寿命的影响也可以忽略.

5 结论与展望

针对视频监控、CDP、VTL、备份、归档等连续数据存储应用,本文提出一种高效能盘阵——Ripple-RAID,

继承了 S-RAID 5 的局部并行思想,设计了新的数据布局,综合运用了地址转换、异地更新、基于流水技术渐进生成校验、分段数据恢复等策略,在单盘容错的前提下,既保持了局部并行的节能性,又有效消除了局部并行带来的小写问题.实验结果表明,Ripple-RAID 具有突出的写性能和节能效率.连续数据存储中的读操作以数据回放为主,因此 Ripple-RAID 一般具有与写性能接近的、突出的读性能.

与 S-RAID 5 显著不同,Ripple-RAID 对写数据的连续性没有要求,其地址映射机制能够把非连续数据映射为连续数据.对于非连续数据存储,如果优化后(如采用分层或混合存储)随机读操作频率很低,较少进行改写操作,则 Ripple-RAID 也是适用的.对 Ripple-RAID 可进一步分类与拓展:把上述采用类似 RAID 5 的分布校验、单盘容错的 Ripple-RAID,称为 Ripple-RAID 5;把采用类似 RAID 4 的集中校验、单盘容错的 Ripple-RAID,称为 Ripple-RAID 4;把采用类似 RAID 6 的分布校验、双盘容错(*P*+*Q* 校验)的 Ripple-RAID,称为 Ripple-RAID 6.

References:

[1] Li X, Tan YA, Sun ZZ. Semi-RAID: A reliable energy-aware RAID data layout for sequential data access. In: Proc. of the 27th IEEE Symp. on Massive Storage Systems and Technologies (MSST). Washington: IEEE Computer Society, 2011. 1−11. [doi: 10.1109/MSST.2011.5937222]

[2] Li YZ, Sun ZZ, Ma ZM, Zheng J, Tan YA. S-RAID 5: An energy-saving RAID for sequential access based applications. Chinese Journal of Computers, 2013,36(6):1290−1301 (in Chinese with English abstract).

[3] Stodolsky D, Gibson G, Holland M. Parity logging: Overcoming the small write problem in redundant disk arrays. In: Proc. of the 20th Annual Int'l Symp. on Computer Architecture (ISCA). New York: ACM Press, 1993. 64−75. [doi: 10.1145/173682.165143]

[4] Menon J, Roche J, Kasson J. Floating parity and data disk arrays. Journal of Parallel and Distributed Computing, 1993,17(1): 129−139. [doi: 10.1006/jpdc.1993.1011]

[5] Jin C, Feng D, Jiang H, Tian L. RAID6L: A log-assisted RAID6 storage architecture with improved write performance. In: Proc. of the 27th Symp. on Mass Storage Systems and Technologies (MSST). Washington: IEEE Computer Society, 2011. 1−6. [doi: 10.1109/MSST.2011.5937230]

[6] Li MQ, Shu JW. DACO: A high-performance disk architecture designed specially for large-scale erasure-coded storage systems. IEEE Trans. on Computers, 2010,59(10):1350−1362. [doi: 10.1109/TC.2010.22]

[7] Shen YL, Xu L. Efficient disk I/O characteristics analysis method based on virtual machine technology. Ruan Jian Xue Bao/Journal of Software, 2010,21(4):849−862 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3492.htm [doi: 10.3724/SP.J.1001.2010.03492]

[8] Zikopoulos P, Eaton C. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. New York: McGraw-Hill, 2011. 15−58.

[9] Zhu QB, Chen ZF, Tan L, Zhou YY, Kimberly K, John W. Hibernator: Helping disk arrays sleep through the winter. Operating Systems Review (ACM), 2005,39(5):177−190. [doi: 10.1145/1095809.1095828]

[10] EMC Corporation. Symmetrix 3 000 and 5 000 enterprise storage systems product description guide. 2012. http://www.emc.com/collateral/software/symmetrix3000-and-5000.pdf

[11] Gurumurthi S, Sivasubramaniam A, Kandemir M, Franke H. DRPM: Dynamic speed control for power management in server class disks. In: Proc. of the 30th Int'l Symp. on Computer Architecture. New York: ACM Press, 2003. 169−179. [doi: 10.1145/871656.859638]

[12] Carrera E, Pinheiro E, Bianchini R. Conserving disk energy in network servers. In: Proc. of the 17th Int'l Conf. on Supercomputing (ICS). New York: ACM Press, 2003. 86−97. [doi: 10.1145/782814.782829]

[13] Weddle C, Oldham M, Qian J, Wang AA, Reiher P, Kuenning G. PARAID: A gear-shifting power-aware RAID. In: Proc. of the 5th USENIX Conf. on File and Storage Technologies (FAST). Berkeley: USENIX Association, 2007. 245−260.

[14] Pinheiro E, Bianchini R. Energy conservation techniques for disk array-based servers. In: Proc. of the 18th Int'l Conf. on Supercomputing (ICS). New York: ACM Press, 2004. 68−78. [doi: 10.1145/1006209.1006220]

[15] Colarelli D, Grunwald D. Massive arrays of idle disks for storage archives. In: Proc. of the ACM/IEEE Conf. on Supercomputing. Los Alamitos: IEEE Computer Society, 2002. 1−11. [doi: 10.1109/SC.2002.10058]

[16] Narayanan D, Donnelly A, Rowstron A. Write off-loading: Practical power management for enterprise storage. In: Proc. of the 6th USENIX Conf. on File and Storage Technologies (FAST). Berkeley: USENIX Association, 2008. 253−267. [doi: 10.1145/1416944.1416949]

[17] Storer M, Greenan K, Miller E, Voruganti K. Pergamum: Replacing tape with energy efficient, reliable, disk-based archival storage. In: Proc. of the 6th USENIX Conf. on File and Storage Technologies (FAST). Berkeley: USENIX Association, 2008. 1−16.

[18] Li D, Wang J. EERAID: Energy-Efficient redundant and inexpensive disk array. In: Proc. of the 11th ACM SIGOPS European Workshop. New York: ACM Press, 2004. 1−14. [doi: 10.1145/1133572.1133577]

[19] Wang J, Zhu H, Li D. eRAID: Conserving energy in conventional disk-based RAID system. IEEE Trans. on Computers, 2008,57(4):359−374. [doi: 10.1109/TC.2007.70821]

[20] Mao B. Research on data layout technologies for disk arrays [Ph.D. Thesis]. Wuhan: Huazhong University of Science & Technology, 2010 (in Chinese with English abstract).

[21] Narayanan D, Thereska E, Donnelly A, Elnikety S, Rowstron A. Migrating server storage to SSDs: Analysis of tradeoffs. In: Proc. of the 4th ACM European Conf. on Computer Systems. New York: ACM Press, 2009. 145−158. [doi: 10.1145/1519065.1519081]

[22] Deng YH. What is the future of disk drives, death or rebirth. ACM Computing Surveys, 2011,43(3):23−49. [doi: 10.1145/1922649.1922660]

[23] Chen F, Koufaty DA, Zhang X. Hystor: Making the best use of solid state drives in high performance storage systems. In: Proc. of the Int'l Conf. on Supercomputing (ICS). New York: ACM Press, 2011. 22−32. [doi: 10.1145/1995896.1995902]

[24] Guerra J, Pucha H, Glider J, Wendy B, Raju R. Cost effective storage using extent based dynamic tiering. In: Proc. of the 9th USENIX Conf. on File and Storage Technologies (FAST). Berkeley: USENIX Association, 2011. 273−286.

[25] Guerra J, Belluomini W, Glider J, Gupta K, Pucha H. Energy proportionality for storage: Impact and feasibility. ACM SIGOPS Operating Systems Review, 2010,44(1):35−39. [doi: 10.1145/1740390.1740399]

[26] The Storage Networking Industry Association. The 2012 SNIA dictionary. 2012. http://www.snia.org/education/dictionary

[27] Rosenblum M, Ousterhout JK. The design and implementation of a log-structured file system. ACM Trans. on Computer Systems, 1992,10(1):26−52. [doi: 10.1145/146941.146943]

[28] Hartman JH, Ousterhout JK. The zebra striped network file system. ACM Trans. on Compute Systems, 1995,13(3):274−310. [doi: 10.1145/210126.210131]

[29] Nippon Technology and Telephone Corporation. NILFS2. 2012. http://www.nilfs.org/en/download.html

[30] Wilkes J, Golding R, Staelin C, Sullivan S. The HP AutoRAID hierarchical storage system. ACM Trans. on Computer Systems, 1996,14(1):108−136. [doi: 10.1145/225535.225539]

附中文参考文献:

[2] 李元章,孙志卓,马忠梅,郑军,谭毓安.S-RAID 5:一种适用于顺序数据访问的节能磁盘阵列.计算机学报,2013,36(6):1290−1301.

[7] 沈玉良,许鲁.一种基于虚拟机的高效磁盘 I/O 特征分析方法.软件学报,2010,21(4):849−862. http://www.jos.org.cn/1000-9825/3492.htm [doi: 10.3724/SP.J.1001.2010.03492]

[20] 毛波.盘阵列的数据布局技术研究[博士学位论文].武汉:华中科技大学,2010.

孙志卓(1973－),男,吉林伊通人,博士,副教授,CCF 会员,主要研究领域为计算机体系结构,网络存储,嵌入式系统.

谭毓安(1972－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络存储,信息安全,嵌入式系统.

张全新(1974－),男,博士,讲师,CCF 会员,主要研究领域为存储算法,移动计算.

李元章(1978－),男,博士生,讲师,CCF 会员,主要研究领域为网络存储,嵌入式系统.