文章编号:1671-4598(2021)12-0150-06 DOI:10.16526/j.cnki.11—4762/tp.2021.12.028 中图分类号:TP316 文献标识码:A

基于虚拟存储层的缓存控制软件设计

贾德明，朱明胜，古 妍
(山东鲸鲨信息技术有限公司，山东 泰安 271000)

摘要：缓存加速技术可以利用固态硬盘（SSD，solid state disk）随机访问性能高的优势，提升机械硬盘的随机读写性能；传统的缓存加速技术难以适应大数据背景下高并发、间歇性频繁访问等热点数据访问需求；为了提升缓存整体性能，提出一种基于虚拟存储层的缓存策略（CVSL，cache policy based on the virtual storage layer），将缓存技术和分层存储技术相结合，通过热度统计、数据逻辑迁移，实现基于数据逻辑分层的缓存控制；实验结果表明，相对传统的缓存策略，CVSL 策略的随机读写性能提升了 9%～10%，未见明显波动，在缓存命中率方面具有良好的效果，达到了预期设计目标。

关键词：缓存加速；自动分层；控制策略；数据迁移；固态硬盘

Design of Cache Control Software Based on Virtual Storage Layer

ZHU Mingsheng, JIA Deming, GU Yan
(Shandong Estor Information Technology Co., Ltd., Tai'an 271000, China)

Abstract: Cache acceleration technology can take advantage of the high performance of random access of solid state drives (solid state disks, SSD) to improve the random read and write performance of mechanical hard drives. Traditional cache acceleration technology is difficult to adapt to the needs of hot data access such as high concurgency and intermittent frequent access under the background of big data. In order to improve the overall performance of the cache, a cache strategy based on the virtual storage layer (The cache strategy basedon the virtual storage layer, CVSL) is proposed, which combines the caching technology with the hierarchical storage technology to realize the cache control based on the data logic layering through thermal statistics and data logical migration. Experimental results show that compared with the traditional caching strategy, the random reading and writing performance of CVSL strategy has improved by 9%～10%, no obvious fluctuations have been seen, and the cache hit rate has a good effect, and the expected design target has been achieved.

Keywords: cache acceleration; automatic tiering; control strategy; data migration; SSD

0 引言

随着 5G、云计算、大数据、物联网等信息技术的发展，数据呈爆炸式增长，对存储系统的海量数据存储能力提出了高性能、多样化的访问要求[1]。针对数据访问局部热点的特性，通常将高频访问的数据缓存在高速存储介质上，以加快数据读写速度[2]。缓存控制策略是缓存系统的核心，决定了缓存空间利用的有效性[3]。传统的缓存策略都是基于访问时间或访问频率进行设计[4-5]。提高缓存命中率可以提高应用程序的运行速度，这是提高缓存性能的关键[4]。基于访问时间仅考虑了时间局部性，忽略历史上的高频访问数据；基于访问频率仅考虑高频访问数据，忽略最近访问的数据可能成为访问热点[6]。两种策略均不能有效满足现有热点数据缓存需求。

缓存数据生命周期控制的核心是根据对数据价值的精准判断，然后根据数据价值将数据分配到对应的、性能合适的存储设备上[7]。据研究结果表明，存储系统中的数据 70%～80%是静止不动的，数据写入后很少被访问。经常访问的热点数据占 20%～30%[8]。其中间歇性热点数据的假性降温表现，可能会从高速缓存介质中移除，一旦被移除，数据块可能因为缓存中暂时性高频访问数据而很难再次迁入缓存层，而导致后面访问持续性延迟过高，影响上层应用运行效率。

在较为复杂的数据访问场景中，传统基于访问时间和基于访问频次的策略对于热点数据的预测略显不足，如图 1 所示。

假设存在缓存块 D1、D2、D3，基于最近访问时间排序，优先级为 D2，D3，D1；基于访问频率排序，优先级为 D1，D2，D3；在基于最近访问时间的策略中，D1 将被淘汰，但 D1 会频繁访问数据，只是最近一段时间没有被访问，因此 D3 替换将来可能被访问的 D1；在基于访问频率的策略中，未考虑访问频率的动态性，导致 D1 和 D2 的访

收稿日期:2021-11-03； **修回日期**:2021-11-15。
基金项目:山东省重大科技创新工程项目(2019JZZY020130)。
作者简介:贾德明(1985-)，男，山东德州人，大学本科，主要从事分布式海量数据存储方向的研究。
通讯作者:朱明胜(1981-)，男，天津人，大学本科，主要从事大数据存储、安全存储方向的研究。
引用格式:贾德明，朱明胜，古 妍. 基于虚拟存储层的缓存控制软件设计[J]. 计算机测量与控制，2021，29(12)：150-155.

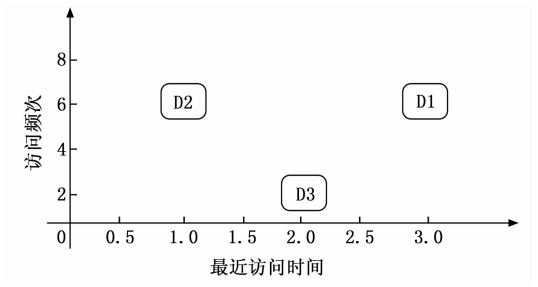

图 1　访问频次访问时间示意图

问频率相同，事实上最近访问的高频数据 D2 再次被访问的可能性更大，不符合实际访问规律[9]。CVSL 策略综合考虑最近访问时间和访问频率，优先级排序为 D2 ＞D1 ＞D3，能够反映数据访问频率随时间变化的变化，更能适应热点数据的访问特点。因此 CVSL 策略克服了基于最近访问时间和基于访问频率缓存策略的不足。

自动分层的数据迁移是定时执行的，短时间内无法快速响应热数据的变化[10-13]。例如，当出现新的热数据时，缓存在第一次访问时就会将其调入 SSD，接下来的读请求都能直接从 SSD 访问；而分层架构，必须等待一段时间（数据迁移涉及不同层之间空间交换，必须后台执行），才会将新的热数据迁移到 SSD，那么在这段时间内，是无法享受到 SSD 的高性能的，甚至在极端的情况下，可能该数据被迁移到 SSD 后，热度已过。另一方面，分层的优点是其有足够的时间更准确地识别冷热数据（识别顺序 IO）。因此，缓存适合负载变化迅速的场景；分层适合静态的负载（冷热数据很少变化）。

此外，固态硬盘由于写前擦的硬件特性，非对齐访问会导致写放大，不能充分发挥固态盘性能优势，而且降低了 SSD 使用寿命[14-19]。

基于上述问题，研究设计了一种基于虚拟存储层的缓存控制策略，该策略引入了虚拟存储层的概念，综合缓存技术和自动分层技术的优缺点，CVSL 在结合访问时间策略和访问频率策略基础之上，引入虚拟存储层概念，有效甄别数据访问热度，避免因间歇性访问的假性冷数据被移除缓存，提高整体缓存命中率。

结合分层存储中热数据、温数据、冷数据的概念，虚拟存储层承载访问频次的统计，充分考虑间歇性局部热点数据的访问特点，尽量避免热点数据因暂时冷却被迁入低速存储介质，将热数据直接降级为冷数据，同时对顺序写 IO 不缓存，再没有增加硬件成本的情况下，提高了缓存命中率，充分发挥固态硬盘优势，提升了综合访问性能。

1　缓存控制的结构及原理

虚拟存储层是通过软件划分出的一个虚拟存储域，逻辑上是位于高速存储介质和低速存储介质之间，存储温数据。对于迁入虚拟存储层的数据块，仅缓存其块描述信息，对应数据实际存储在低速介质中。虚拟存储层不存储实际数据，数据会直接存储在低速存储介质，但在虚拟存储层中保留数据块的访问时间，访问频次等信息。在虚拟存储层中的数据会根据访问热度决定数据是重新迁入高速存储介质还是迁入低速存储介质。其中，数据块在虚拟存储层和低速存储介质之间流动，只需要移除该缓存块的块描述信息据即可，无须数据迁移，不会增加低速存储介质额外的 IO 负担。

CVSL 缓存控制逻辑架构如图 2 所示。

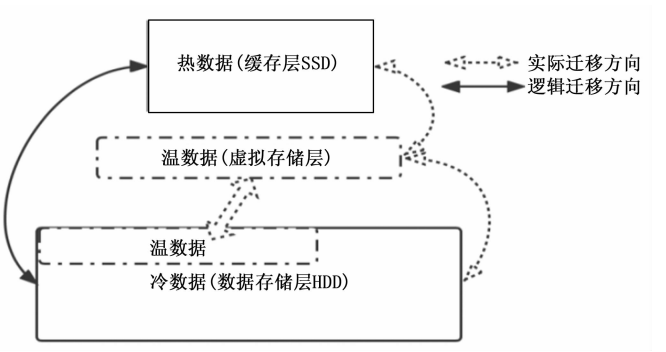

图 2　CVSL 缓存控制逻辑架构

数据物理从高速存储介质（SSD）和低速存储介质（HDD）之间迁移，逻辑上从高速存储介质、虚拟存储层、低速存储介质之间迁移。温数据和冷数据均存放在低速存储介质 HDD 上。

CVSL 缓存系统由以下模块组成，如图 3 所示。

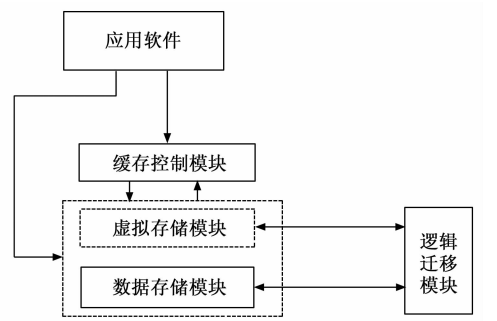

图 3　CVSL 模块组成

1）缓存控制模块，主要负责缓存数据的基本控制功能，按照 SSD 物理特性设计的，根据 SSD 擦除块以 chunk 为最小单位（默认 512K）分配 SSD 空间，使用 B＋树、最近最少使用（LRU，least recently used）链表、最不经常使用（LFU，least frequently used）链表混合方法来跟踪缓存数据，缓存数据可以是 chunk 上的任意一个扇区。最大程度上减少随机写的代价。定期唤醒或者达到空间阈值上限回写缓存脏数据。

缓存页控制根据 SSD 擦除块大小将 SSD 空间分成若干个 chunk。缓存数据与元数据都是按 chunk 来组织的。

（1）分配器：COW（copyonwrite）式的空间分配，分配的单元是 chunk，数据在 chunk 内部全部都是追加写入

的，不会出现覆盖写，当有覆盖写时，会重定向到新的数据块。

（2）元数据部分：B+树节点数据是最主要的元数据，也是 COW 式分配，对于 B+树节点的修改，需要先分配新的节点，将新数据写入，再丢弃老的节点。

2）逻辑迁移模块，主要负责根据数据块的访问时间访问次数信息进行热度统计，根据统计结果在虚拟存储层和数据存储层之间进行逻辑调度。

3）虚拟存储模块，主要负责维护迁入虚拟存储层数据块的热度信息以及与数据存储层的映射关系。逻辑上使用 page（4k）为最小单位管理数据块。虚拟存储层在内存中开辟一定空间，暂存待迁入数据块，多个数据块合并后按照 chunk 对齐迁入，降低 SSD 写入次数，尽量避免数据抖动影响性能。

4）数据存储模块，主要负责实际存储所有的数据块，包括虚拟存储层的数据块，因为数据存储层一般容量较大，所以采用 1 M 为最小单位管理数据块，后面可设计为可变模式，支持 1～64 M 可调节。

数据存储格式如图 4 所示。

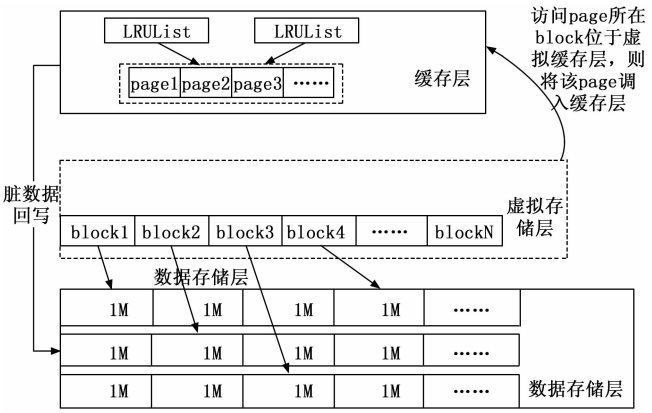

图 4 数据存储格式

相比传统策略，CVSL 综合考虑了访问时间策略和访问频率策略，兼顾了 SSD 的物理特点，并且缓存块在移除缓存层时，其访问时间访问频次等信息仍维护在虚拟存储层和数据存储层中。数据热度信息在数据块整个生命周期中均有效，有助于缓存层、虚拟存储层和数据存储层进行热点数据追踪。

2 缓存控制软件设计

CVSL 通过缓存页管理将热点数据存储在高速介质上，虚拟存储层位于 IO 缓存层和数据存储层之间，将温数据和冷数据逻辑隔离，防止冷热数据频繁迁入迁出造成抖动，造成性能波动，提高热点数据命中率。

2.1 缓存控制设计

缓存控制模块采用 B+树索引管理 HDD 和缓存数据的映射关系，建立 LRU 和 LFU 两个链表，根据访问时间和访问次数控制缓存块的迁入迁出。

当有上层应用访问未在缓存中，但对应 block 在虚拟存储层时，判定该缓存页为热点数据，调入缓存层，根据 LRU 和 LFU 链表移除最长时间没有使用以及最近最不常用的缓存页。

当有上层应用访问为在缓存中，且对应 block 未在虚拟存储层时，暂不将数据块迁入缓存层。

被移除的缓存页由调度模块管理，根据其数据热度迁入虚拟存储层，或者数据存储层。如果缓存页为非脏数据，该过程为逻辑分层，仅涉及元数据修改，不会对机械盘进行读写操作。

缓存控制模块还承载着脏数据回写的功能。当缓存层中脏数据达到一定阈值会主动触发回写线程进行脏数据刷写；内置定时器会触发回写线，定时扫描脏数据块进行脏数据回写。

脏数据回写时不会更新数据块访问次数、访问时间等信息，直接写入数据存储层，防止内部数据刷写影响虚拟存储层的数据热度评估，造成缓存数据块热度出现偏差；

数据在写入数据存储时，合并排序写入，充分利用机械盘的特性，提高数据写入效率。

上层请求到达时，如果是顺序 IO，则不进行缓存，直接绕过缓存控制层；如果数据块在 SSD 缓存层，则直接读写访问并向上返回请求结果；如果缓存未命中，则访问虚拟存储层。

如果数据在虚拟存储层，则根据映射关系在数据存储层访问数据，向上返回请求结果，并且将该数据块调入缓存层（此处会尝试多个缓存页合并调入缓存层，减少 SSD 写入次数）。

如果未在虚拟存储层，则继续向数据存储层请求，未再虚拟存储层的数据块只更新访问次数信息，不立即调入缓存层。

访问完毕后，更新内存中对应 block 的访问时间、读写次数等信息。

由于虚拟存储层仅存储缓存块的块描述信息，使用较小的空间即可存储较大范围的块描述信息，即可在较大范围内筛选间歇性热点数据，有效提高了此类数据的缓存命中率。

IO 缓存层最大程度上保护数据，在系统异常关机时数据仍然是可靠的。因为只有在数据完全写回存储设备才确认写成功。SSD 的特点就是随机 IO 性能很高，而对于顺序 IO 提升却并不大，所以会检测顺序 IO 并透传给数据存储层。

访问流程如图 5 所示。

2.2 虚拟存储层热度统计

热度统计线程会周期性遍历设备所有的 block，block 的热度由 4 个参数衡量，包括最近写时间、被写次数、最近读时间、被读次数。

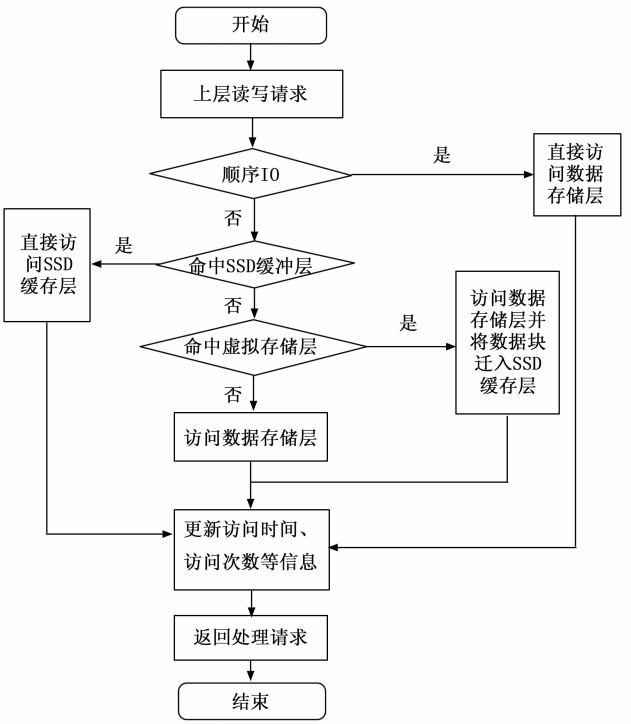

图 5　应用访问流程图

逻辑上升的 block 满足：1）自上次统计以来，被写次数大于 W_{hicount}；或者，2）自上次统计以来，被读次数大于 R_{hicount}。满足条件的 block 按被访问时间和被访问次数降序排列，热数据块在前，形成逻辑上升数据块链表 $Queue_{\text{up}}$。

逻辑下降的 block 满足：1）自上次统计以来被写次数小于 W_{lowcount}；并且，2）自上次统计以来被读次数小于 R_{lowcount}。满足条件的 block 按被访问时间和被访问次数升序排列，冷数据块在前，生成逻辑下降数据链表 $Queue_{\text{down}}$。

W_{hicount}为写次数高水位线，R_{hicount}为读次数高水位线，W_{lowcount}为写次数低水位线，R_{lowcount}，为读次数低水位线，可以根据实际业务调整，注意需满足 $W_{\text{hicount}} \geqslant W_{\text{lowcount}}$，$R_{\text{hicount}} \geqslant R_{\text{lowcount}}$，默认 $W_{\text{hicount}} = W_{\text{lowcount}} = R_{\text{hicount}} = R_{\text{lowcount}} = AVG_{\text{rwcount}}$（平均读写次数）

热度统计完毕后，热度统计线程会清除数据块中的访问时间、读写次数信息。热度统计流程如图 6 所示。

2.3　虚拟存储层逻辑迁移

在确定了需要向上或者向下迁移的 block 后，数据迁移线程将待迁移的 block 逐个迁移。在迁移每个 block 前会判断是否停止此次迁移任务。停止的条件有：1）迁移线程工作时间限制，如果超时就中止这次迁移；2）检查目前虚拟存储层的逻辑利用率，如果利用率为大于 95%，则只允许向下迁移任务；如果利用率小于 60%，则只允许执行向上迁移任务；如果利用率大于 60%小于 90%时，则同时允许执行向上迁移和向下迁移任务。

该数据迁移不会对数据块进行读写，仅修改虚拟存储层的逻辑关系，不会对机械盘造成额外的性能负担。由于缓存层管理单位为 page＝4 k，数据存储层管理单位为 block＝1 M。所以当命中虚拟存储层 block 时，仅将其访问的 page 调入缓存层，不需要加载整个 block。

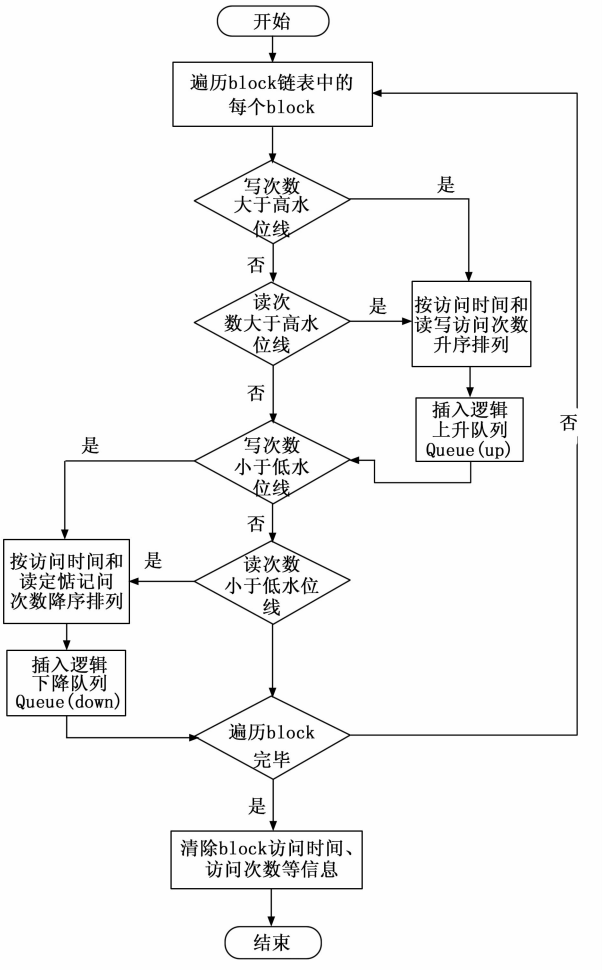

图 6　热度统计流程图

逻辑迁移流程如图 7 所示。

由于写入数据存在热数据：温数据：冷数据＝15%：5%：80%的特点，建议 SSD：HDD 容量比为（15＋5）%：80%。其虚拟存储层 block 总数计算公式如下：

$$C_{\text{block}} = V_{\text{hdd}} \times (15\% + 5\%) \div S_{\text{block}} \tag{1}$$

式中，C_{block}代表虚拟存储层管理的 block 总数，V_{hdd}代表磁盘容量（单位 MiB），S_{block}代表 block 大小（单位 MiB）。

3　实验结果与分析

为了验证 CVSL 中基于虚拟存储层设计的缓存管理和动态迁移策略的有效性，本节采用具有代表性的 Linux 开源缓存加速软件 dmcache 和 CVSL 模块进行了随机 IO 性能对比测试实验。

本实验基于 FT2000＋平台和 Linux 操作系统，dmcache 模块（对应内核版本 4.19），50GSSD＋1 片 1TSATA 机械硬盘进行测试。测试工具采用 fio－3.28，direct 访问，

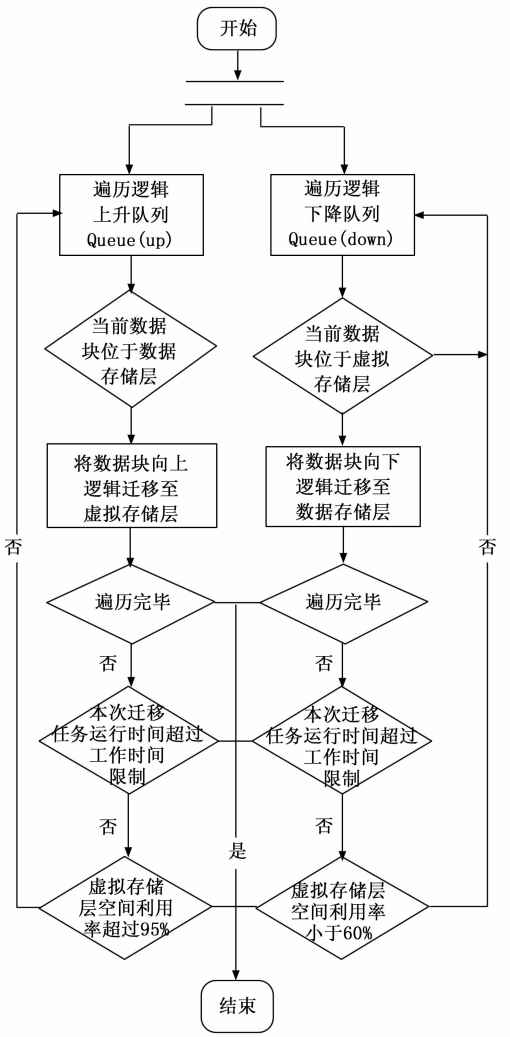

图 7 数据逻辑迁移流程图

排除服务器内存影响。

首先使用 fio 分别测试 SSD 和 HDD 的 4K 随机读写性能，测试 5 min，确保 SSD 和 HDD 机械盘性能表现正常。

性能测试流程具体如下：

步骤一：SSD 全盘 trim 后，使用 dmcache 模块将 SSD 绑定为 HDD 的缓存设备。

步骤二：使用 fio 进行随机 4 K 随机读写预热模拟应用访问，读写比例为 4∶6，预热时间为 48 h。

步骤三：预热完毕后，使用 fio 分别进行 4 K 随机 IO 读写测试，测试 5 分钟，记录测试结果。

步骤四：重复步骤二和步骤三 3 次，并记录测试结果。

CVSL 模块测试方法同步骤一～步骤四。

1）4 k 随机写性能测试结果对比，如图 8 所示。

dmcache 的 4 次随机写测试结果分别为 781，712，799，752，平均值为 761；CVSL 的 4 次随机写测试结果分别为 840，835，846，841，平均值为 840，相对 dmcache 提升约 10.5%。

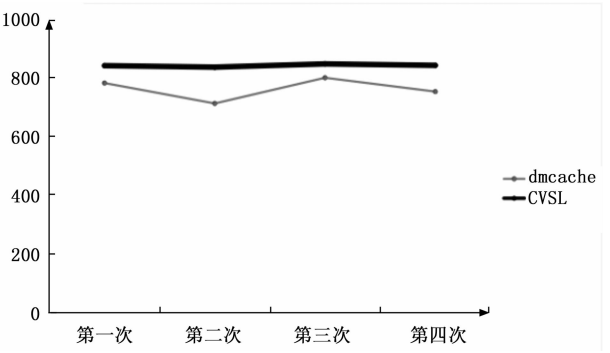

图 8 四次 4 k 随机写测试结果

2）4 k 随机读性能测试结果对比，如图 9 所示。

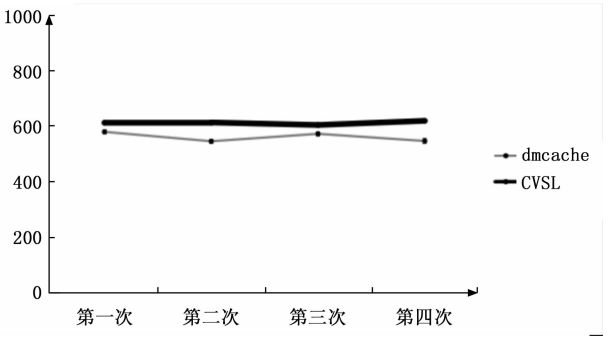

图 9 四次 4 k 随机读测试结果

dmcache 的 4 次随机读测试结果分别为 587，553，580，554，平均值为 569；CVSL 的 4 次随机读测试结果分别为 620，621，612，627，平均值为 620，相对 dmcache 提升约 9%。

从图 8 和图 9 可以看出，CVSL 相比 dmcache 模块，随机读写性能均有明显提高，缓存加速性能提高了 9%～10%。而且 CVSL 在进行了 4 次 48 小时的模拟应用访问后，4 k 随机写 IOPS 稳定在 840 左右，4 k 随机读 IOPS 稳定在 620 左右，性能值比较平稳。从实验结果分析，CVSL 在未增加硬件成本的情况下，有效的提高了机械硬盘的性能，而且在模拟间歇性数据访问后，性能没有明显波动，达到了预期设计效果。

4 结束语

在全球数据爆炸性增长的背景下，如何结合固态硬盘和传统机械硬盘的优势应对海量数据的高效存储，具有很高的社会意义和研究价值。针对缓存数据尤其是间歇性频繁访问的缓存数据特点，引入虚拟存储层概念，在传统缓存策略上融合了自动分层技术并做了改进，主要有以下技术优势：1）综合考虑了访问时间和访问频率对缓存块的影响，结合了传统缓存适合负载变化迅速的场景自动分层适合静态负载的特点，对热数据判定更精准。充分利用自动分层热点数据升温和降温较慢，可更准备识别热点数据的技术原理，降低间歇性热点数据变为冷数据的概率，也降低了缓存页抖动对性能造成的影响；2）虚拟存储层和

数据存储层之间迁移数据时，仅需修改数据块描述信息和映射关系。缓存层淘汰非脏缓存页时，也是仅做页面信息修改，都不会没有对机械硬盘进行读写操作，不会增加额外的磁盘 IO，对访问性能没有影响；3）虚拟存储层不存储真实数据，可在较大范围内承载因暂时不访问被移出缓存层的间歇性热点数据，有效的提高了缓存命中率；4）数据块的访问时间访问频次具有全局性，提高了热点数据追踪效率；数据存储层使用 block 为最小单位，降低了元数据的占用空间。相比传统缓存策略，在没有增加硬件成本的情况下，降低了缓存页被误判为冷数据的概率，提高了缓存命中率。

基于虚拟存储层的 SSD 缓存控制策略可应用于磁盘阵列、分布式存储等企业级网络存储系统，能够有效提高在云平台、大数据分析、数据库等高并发随机读写应用场景下的存储访问性能，提高应用系统的响应时间。

参考文献：

[1] 罗　乐，刘　轶，钱德沛. 内存计算技术研究综述 [J]. 软件学报，2016，27 (8)：2147－2167.

[2] 葛　微. 大数据索引和查询优化技术与系统研究 [D]. 南京：南京大学，2018.

[3] 莫云明. 基于内容流行度和节点中心度的协同缓存策略 [D]. 武汉：华中科技大学，2019.

[4] 张　远. 一种基于缓存分区的线程间缓存干扰消除结构的设计和实现 [D]. 南京：东南大学，2016.

[5] 孙文庆，郁文清. 超融合分布式存储架构在中移互联网资源池中的应用 [J]. 移动通信，2018，42 (7)：29－33，38.

[6] 潘　晓，马　昂，等. 基于时间序列的轨迹数据相似性度量方法研究及应用综述 [J]. 燕山大学学报，2019，43 (6)：531－545.

[7] 徐尔茨. 固态硬盘存储系统关键技术研究 [D]. 长沙：国防科技大学，2019.

[8] 周　斌，汪　浪，张　莹，等. 基于数据块级迁移策略的设计与实现 [J]. 计算机工程与设计，2016，37 (7)：1822－1826.

[9] 冯东煜，王鹏达，黄飞龙，等. 一种基于数据热度的缓存策略 [J]. 信息与电脑（理论版），2021：33 (12)：196－199.

[10] 董新华，李瑞轩，等. Hadoop 系统性能优化与功能增强综述 [J]. 计算机研究与发展，2013，50 (s2)：1－15.

[11] 潘　巍，李战怀，等. 新型非易失存储环境下事务型数据管理技术研究 [J]. 软件学报，2017，28 (1)：59－83.

[12] 鲍　禹，付印金，陈卫卫. 多云存储关键技术研究进展 [J]. 计算机工程，2020，46 (10)：18－32，40.

[13] 屠雪真. 分布式缓存系统客户端关键技术研究 [J]. 电脑编程技巧与维护，2019 (4)：4－9，12.

[14] 陆游游，舒继武. 闪存存储系统综述 [J]. 计算机研究与发展，2013，50 (1)：49－59.

[15] 金培权，郝行军，岳丽华. 面向新型存储的大数据存储架构与核心算法综述 [J]. 计算机工程与科学，2013，35 (10)：12－24.

[16] 方才华，刘景宁，童　薇，等. 全程优化的固态硬盘垃圾回收方法 [J]. 计算机应用，2017，37 (5)：1257－1262，1281.

[17] 石　伟，汪东升. 基于非易失存储器的事务存储系统综述 [J]. 计算机研究与发展，2016，53 (2)：399－415.

[18] 杨元华. 固态盘存储系统性能优化研究 [D]. 武汉：武汉大学，2019.

[19] 黄　赛. 基于闪存的缓存管理研究 [D]. 长沙：华中科技大学，2016.

（上接第 149 页）

[2] 马万磊，戴　滨，蒋　寒. 基于自主诊断重构技术的航天器故障检测系统设计 [J]. 计算机测量与控制，2021，29 (9)：5－9，22.

[3] LUO Y，LU Z. Feasibility analysis for application of domestic neokylin operating system in radar measurement and control system [C] //2017 2nd International Conference on Computer Engineering，Information Science and Internet Technology，2017：526－529.

[4] XU H Y，GAO Y，LIU K M，et al. Research on cross－communication based on real－time Ethernet POWERLINK [C] //2014 26th Chinese Control And Decision Conference (CCDC)，2014：2577－2581.

[5] WEI，et al. Analysis and testing NeoKylin's clock system [C]. 2014 International Conference on Simulation and Modeling Methodologies，Technologies and Applications (SMTA 2014)，2014：47－54.

[6] 马　威，姚静波，常永胜，等. 国产 CPU 发展的现状与展望 [J]. 集成电路应用，2019，36 (4)：5－8.

[7] 綦志勇，常排排. 国产操作系统的机遇与挑战 [J]. 电脑知识与技术，2020，16 (14)：252－253.

[8] 徐明远. 基于 VxWorks 的雷达自动测试系统软件开发 [D]. 哈尔滨：哈尔滨工业大学，2015.

[9] 刘康丽，谷　静，杜　影，等. 中标麒麟系统下基于 QT 的 PXI 仪器软件开发 [J]. 计算机测量与控制，2019，27 (10)：159－163.

[10] 姜家文，许荣胜，胡振宇. 中标麒麟环境下基于 Qt 的神通数据库编程浅析 [J]. 软件工程，2017，20 (3)：18－20.

[11] 中标软件有限公司. 中标麒麟安全云操作系统解决方案 [J]. 信息技术与标准化，2012，12：41－44.

[12] 陆文周. Qt5 开发及实例 [M]. 北京：电子工业出版社，2014.

[13] LIU C L，LAYLAND J W. Scheduling algorithmsfor multiprogramming in a hard－real－timeenvironment [J]. Journal of the ACM (JACM)，1973，20 (1)：46－61.

[14] 崔　健. 基于中标麒麟的火箭飞行三维实时仿真系统的研究与实现 [D]. 重庆：重庆大学，2014.