计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140019
52(7):1558-1566, 2015

面向新型非易失存储器的文件级磨损均衡机制

蔡 涛[1] 张永春[1] 牛德姣[1] 倪晓蓉[1] 梁东莺[2]
[1](江苏大学计算机科学与通信工程学院 江苏镇江 212013)
[2](深圳信息职业技术学院信息中心 广东深圳 518055)
(caitao@ujs.edu.cn)

File System Level Wear Leveling Mechanism for Non-Volatile Memory Based Storage

Cai Tao[1], Zhang Yongchun[1], Niu Dejiao[1], Ni Xiaorong[1], and Liang Dongying[2]
[1](*School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu* 212013)
[2](*Information Center, Shenzhen Institute of Information Technology, Shenzhen, Guangdong* 518055)

Abstract The access speed of STTRAM, MRAM and the other Non-volatile memory is close to that of DRAM. So they are very useful for high performance storage system and improving the performance of large computer system, but their limited write endurance is one of the most important limitations. We introduce file system level wear leveling technology for them. Using the Hash function to disperse files on the storage system, some blocks are avoided allocating repeatedly when creating and deleting files. The blocks with lower write count are chosen to avoid some blocks with centralized write operation when allocating space for file. An active heat data migration strategy is designed to reduce the I/O performance impaction of wear leveling mechanism. Finally, we implement the file system level wear leveling mechanism prototype based on the open source object-based storage device named Open-osd. Using Filebench, postmark and some trace are tested and analyzed, and the results show that the difference of write count between blocks is reduced to around one of the twentieth with original, and only reducing 6% system I/O performance and increasing about 0.5% amount of writing. It is verified that the file system level wear leveling mechanism is effective and stable.

Key words non-volatile memory; wear leveling; file system; storage system; new memory

摘 要 自旋转移力矩磁存储器(spin transfer torque random access memory, STTRAM)和磁阻式随机存储器(magnetic random access memory, MRAM)等新型存储器具有接近于DRAM的访问速度,是构建高性能外存系统和提高计算机系统性能的重要手段,但有限的写次数是其重要局限之一.设计了文件系统级磨损均衡机制,使用Hash函数分散文件在外存中的存储,避免在创建和删除文件时反复分配某些存储块,通过分配文件空间时选择写次数较低的存储块,避免写操作的集中;使用主动迁移策略,在外存系统I/O负载较低时主动迁移写次数较高的数据块,减少磨损均衡机制对I/O性能的影响.最后在开源的基于对象存储设备Open-osd上实现了面向新型存储器文件系统级磨损均衡机制的原型,使用存储系统通用测试工具filebench和postmark的多个通用数据集进行了测试与分析,验证了基于新型

收稿日期:2014-01-07;**修回日期**:2014-06-04
基金项目:国家自然科学基金项目(61300228);交通运输部信息化技术研究项目(2013364836900);浙江省自然科学基金项目(LY13F020012);广东省自然科学基金项目(S2011010006118)

存储器的文件系统级磨损均衡机制能稳定地将存储块写次数差减少到原来的1/20左右,同时最高仅损失了6%的I/O性能和增加了0.5%的额外写操作,具有高效和稳定的特性.

关键词　非易失存储器;磨损均衡;文件系统;存储系统;新型存储器

中图法分类号　TP316

信息社会的发展使得计算机系统需要处理的信息呈现爆炸式的增长,各种高性能计算机系统是处理海量信息的重要工具.计算机系统在计算能力快速提高的同时,高速的存储系统成为提高计算机系统性能的重要方法.据统计,近40年来计算机的计算能力提高了近20万倍,但同时存储系统的访问延迟只减少了90%、访问带宽只提高了163倍[1],因此I/O性能已成为决定计算机系统性能的重要因素.传统磁介质硬盘需要转动盘片和移动读取臂等机械动作,I/O性能难以提高;基于Flash的固态硬盘(solid state disk, SSD)虽然避免了机械动作,但Flash本身的写寿命很低,很容易造成存储系统的失效,采用异地写、磨损均衡和垃圾回收等机制又会降低I/O性能,并影响I/O性能的稳定;此外Flash与磁盘一样仍然以Page为读写单位,使得小文件的读写性能较低.

以自旋转移力矩磁存储器(spin transfer torque random access memory, STTRAM和磁阻式随机存储器(magnetic random access memory, MRAM)为代表的新型非易失存储器能提供接近DRAM的读写速度并支持以字节为单位的读写方式,但无需刷新操作,相比动态随机存储器(dynamic random access memory, DRAM)具有能耗和密度等多方面的优势,是代替传统硬盘和基于Flash的SSD,是提高外部存储系统I/O性能的有效途径.但新型非易失存储器虽然比Flash的耐写次数高4个以上的数量级,但仍然比传统硬盘低7～8个数量级,依然存在使用寿命的问题.相比Flash,新型非易失存储器能以字节为单位读写数据,这为减少写操作提供了便利,但也增加了磨损均衡和空间管理的难度;同时新型非易失存储器不需要Flash的擦除操作,能直接写入新的数据,这减少了回收垃圾数据的开销,但也增加了存储单元失效的概率;相对Flash常被用于的内存,外存在访问方式、特性等方面均存在很大的差异;直接使用Flash中缓存和磨损均衡等减少写次数、提高使用寿命的机制,难以满足使用新型非易失存储器构建外存系统的要求.新型非易失存储器的读写速度要高于Flash好几个数量级,这在能提供更高I/O性能的同时,也对磨损均衡机制的时间开销提出了更高的要求;因此在使用新型非易失存储器构建外存系统时,高效和性能稳定的磨损均衡机制是一个急需解决的重要问题.

我们在分析外存系统特性的基础上,提出了一种文件系统级磨损均衡机制,改变文件系统中现有的空间分配和管理方式,主动将数据较均匀的分布到外存系统中,避免对存储单元的过度写操作,减少需要迁移的数据,降低触发磨损均衡机制的概率;同时辅助以热数据主动迁移策略,动态避免写操作的集中,提高基于新型非易失存储器外存系统的使用寿命.

1　相关工作

目前关于新型存储器写寿命的研究主要集中于新型存储器作为内存时存在的问题,文献[2]将新型存储器和DRAM混合构建主存,交替分配新型存储器和DRAM,使用DRAM承担大部分写操作,但这无法应用于需要长期保存数据的外存系统;文献[3-4]使用DRAM作为新型存储器的缓存,降低写新型存储器主存的数据量,但DRAM在断电的情况下无法保存数据,无法用于构建外存系统.

现有基于新型存储器外存系统的研究主要集中在如何提高I/O性能方面.文献[5]提出了SCMFS(storage class memory file system),可以不按照块来管理和访问外存系统,同时获得较高的I/O性能,但没有考虑如何延长外存系统的使用寿命.文献[6]提出了BPFS(a file system designed for byte-addressable, persistent memory technologies),使得I/O性能比RAMFS(random rccess memory file system)提高了4～10倍;文献[7]提出了2次写策略,使相变存储器(phase change memory, PCM)介质的I/O性能增加39%,但他们同样缺乏延长新型存储器寿命的机制.文献[8]提出了使用数据压缩策略来降低写数据量从而提高PCM作为嵌入式外存时的使用寿命,但每次读写数据前都要进行解压和压缩会增加额外的时间开销,这将会影响系统的I/O速度,并且增加了CPU的计算量.

文献[9]提出了使用缓存系统来替代嵌入式Flash的磨损均衡机制,缓存无法长久保持数据,存在数据丢失的风险,不适合于解决新型非易失存储器的写寿命问题.

文献[10]设计了CAFTL,在FTL中通过双层映射检测冗余更新数据,删除和合并重复数据,减少写入的数据量;但是该机制需要不断更新地址映射,影响I/O性能,不适合用于对FTL时间开销很敏感的新型非易失存储器管理.文献[11]提出了IPL(in page logging)的机制,在闪存块中保留8 KB大小的log区域保存更新的数据,并使用延迟合并机制,减少了Flash中页的更新次数;而新型非易失存储器支持字节方式的读写操作,不需要以页的方式进行更新,延迟合并机制也会降低基于新型非易失存储器存储系统的I/O性能.文献[12]设计了Delta FTL,使用比较新旧页的方法压缩逻辑页的写数据量,在delta log区域保存多个压缩后的写操作,并合并多个写操作,减少更新Flash中页的次数和数据量;但对数据进行解压和压缩需要较大的时间和空间开销,难以直接应用于对FTL时间开销要求更高的新型非易失存储器管理机制.文献[13-14]中闪存阵列厂商Pure Storage和XtremeIO均通过重复数据删除技术,减少写入闪存阵列的数据量;但新型非易失存储器能支持原地覆盖写操作,所产生的重复数据和对重复数据删除的要求均与Flash有很大区别.文献[15]总结了多种延长闪存存储系统使用寿命的方法,但是新型非易失存储器的写寿命比闪存高出多个数量级,同时能支持原地覆盖写操作,在延长使用寿命的要求方面存在很大的区别;同时新型非易失存储器读写延迟接近于DRAM,因此对管理机制的时间开销更敏感,所以现有延长闪存存储系统使用寿命的方式难以直接用于新型非易失存储器.

文献[16]设计了CASH,利用SSD作为磁盘缓存构建了混合存储系统,在SSD中缓存读操作频繁但写操作较少的数据,以delta的形式将SSD中更新数据的追加到磁盘中,减少对SSD的写操作;但在写频繁的情况下会严重影响存储系统的I/O性能,不适合用于新型非易失存储器的管理.

文献[17]提出了Curling-PCM,是一种针对PCM作为嵌入式系统外存时的磨损均衡策略,把写外存数据量降低了85.7%以上,但是该策略需要对嵌入式应用程序的写操作进行研究和预测;而对面向计算机系统的应用所产生的写操作进行分析和研究就要复杂的多,所以它并不适应于计算机外存系统.

2 文件级磨损均衡机制

新型非易失存储器具有接近于DRAM的读写性能优势,但作为外存时仍然存在写寿命的问题.Flash的研究中常使用DRAM构建缓存,减少对新型存储器的写操作,延长使用寿命;但用于外存系统时,会出现当系统掉电丢失数据的问题.磨损均衡机制能在Flash中页写操作过多后,通过地址重定向的方法,将对相同地址的写操作转移到其他页中,达到增加使用寿命的目的;但在每次读写时都需要查找地址映射表,时间开销较大;Flash中存储数据量的增加也会影响磨损均衡机制的时间开销,同时数据迁移还会带来额外的写操作.

新型非易失存储器的写寿命高于Flash 4～5个数量级,使用寿命的问题远没有Flash那么突出;与内存保存正在使用的数据不同,外存系统用于存储需要长期保存的数据,单个单元的写频率相对内存要低很多;同时因为容量大,空间的使用率也不会太高,因此对均衡各单元写次数的要求没有Flash那么强烈.但新型非易失存储器的读写速度远高于Flash,使得磨损均衡机制对I/O性能的影响更显著,因此要求磨损均衡机制具有时间开销小且稳定的特性.

存储设备管理数据时,只能使用块为单位,无法感知数据的语义,难以针对文件的特性进行优化;文件系统则拥有数据所有的附加信息,管理数据时具有更好的灵活性和优化的能力.我们以文件系统为基础,通过修改文件系统的空间分配机制,预防新型非易失存储器中某个单元的写次数过多,同时保持文件系统仍然具有较高和稳定的I/O性能;同时采用主动迁移策略,防止某些单元写次数较多存在的失效隐患.

2.1 文件均匀分布算法

现有文件系统均是针对传统磁盘设计的,主要优化存储和读取数据时移动磁臂的距离,从而提高I/O性能,未涉及存储介质的写寿命问题;在存储新文件时经常就近使用旧文件释放的区域,在反复删除和创建文件时,很容易造成存储介质某些区域的写次数过多.新型非易失存储器中没有机械部件,存储和读取的速度很快,文件可以随机零散存放,但各存储单元存在写寿命的限制.我们改变文件系统中存储空间的分配机制,增加存储块的写次数等因素,设计文件均匀分布算法.

本文中,avg_count 表示所有存储块的平均写次数,$avg\_count \in \mathbb{N}$,K 表示存储块是否可被分配的系数,$K \in \mathbb{N}$ 且 $0 < K \leqslant 1$,$write\_count$ 表示存储块的写次数,$write\_count \in \mathbb{N}$.

为了在存储空间中均匀分布文件,我们使用 BKDRHash 函数,将文件的路径名作为输入参数计算 Hash 值,并查找存储块的写次数,为文件分配存储空间.具体的算法步骤如下:

算法 1. FileAllocation.

File_allocation(*Filename*,*Size*){
  按照文件路径名 *Filename* 计算 Hash 值;
  While(未分配完成){
    If(该存储块为空闲并且写次数小于 $K \times write\_count$){
      分配该存储块给对应的文件;
      该存储块的写次数 *write_count* 值加 1;
    }
    跳到下一个存储块;
  }
}

文件均匀分布算法能依据文件路径名将文件均匀分布到外存空间中,避免新文件继续使用旧文件的存储空间带来的写操作集中问题;同时记录了每个存储块的写次数,在分配存储块之前检查是否处于当前较低的写次数范围内,避免对同一存储块的大量写操作.文件均匀分布算法中的 avg_count 值并不需要做精确测量,只需要定期进行更新,从而避免对 I/O 性能的影响;此外可以定期或在达到某个阈值后将存储块的 $write\_count$ 和 avg_count 进行相同的减操作,以减少计数所需的存储空间.对于反复删除和创建相同路径名文件的情况,在初期会被分配到同一地址,但随着该存储块写次数的增加,新创建的同名文件会被写到其他存储块中,从而避免了同一存储块的写操作过多.

为了保持较高的 I/O 性能,管理新型非易失存储器时仍然采用大小固定的存储块.在存储大文件时,使用链表存储文件所对应的各存储块,并将链表和元数据等保存到第一个存储块中,便于判断存储块所属的文件和查找文件中数据的地址.修改之后的文件数据读取算法如下:

算法 2. ReadFile.

Read_file(*fileName*,*offset*,*file_len*){
  使用文件路径名 *fileName* 计算 Hash 值;
  使用存储块中保存的元数据判断是否是该文件的首存储块;
  取出存储块链表的首地址;
  根据 *offset* 计算所访问地址在存储块链表中的位置;
  找到所对应的存储块地址;
  访问相应的存储块.
}

存储块可分配系数 K 的值较小时,能更好地选择出写次数较小的存储块进行分配,但过小的 K 值会影响分配空闲数据块的速度;K 的值较大时,能较快找到分配的存储块,但过大的 K 值会使文件均匀分布算法失效.在系统运行时,也可以根据系统要求和存储块写次数的分布情况,动态调整 K 值.

2.2　热数据主动迁移策略

文件均匀分布算法能避免新创建的文件使用被删除文件所在存储块的问题,防止在分配文件时集中在某些存储块中,预防了存储块写操作过多的问题.但文件创建后频繁的写操作仍然能导致某些存储块的写操作过多,这是文件均匀分布算法无法解决的.新型非易失存储的写寿命相比 Flash 提高了 4～5 个数量级,因此写寿命的问题远没有 Flash 那么突出,直接采用 Flash 的相应算法,在损失大量 I/O 性能的同时,又可能增加不必要的一些写操作;新型非易失存储的 I/O 性能远高于 Flash,因此对磨损均衡机制的时间开销更敏感;同时新型非易失存储的写速度又比读速度要慢几个数量级,因此在迁移写次数较多的存储块中数据的同时,减少磨损均衡机制的写操作数量也是一个重要的问题.

依据记录存储块写次数 $write\_count$ 的值,采用主动监测的方法,记录需要迁移的存储块;在 I/O 负载较低的情况下启动离线迁移线程,迁移写操作集中的存储块.

文中用 S 表示判断存储块是否需要置换的系数,$S \in \mathbb{R}$ 且 $0 < S$,T 表示置换线程被激活后因为系统 I/O 负载忙而阻塞的时间,$T \in \mathbb{N}$.

修改文件系统的写操作,在执行写操作之前增加主动监测算法,判断该存储块的写次数是否过多,如果过多则加入离线迁移队列中.主动监测算法如下:

算法 3. DataMonitor.

Data_monitor(*address*){

```
    按照存储块的地址 address 查找该块的写次
      数 write_count;
    If(write_count>S×avg_count){
      将该 address 加入离线迁移队列;
    }
  }
```

存储块置换系数 S 的值较大时,可以减小置换存储块的概率,提高对文件系统 I/O 性能的影响;S 的值较小时,均衡存储块写次数的效果更好,但过小 S 的值会影响文件系统的 I/O 性能.

离线迁移线程在 I/O 负载较低的情况下,选择写次数较少的存储块,迁移离线迁移队列中存储块所对应的数据.

算法 4. OfflineTransfer.

```
Offline_transfer(){
  While(离线迁移队列不为空且 I/O 负载较低)
  {
    If(当前 I/O 负载较低){
      取出离线迁移队列中待转移存储块的地
        址;
      Do{
        查找下一个地址存储块的写次数
          write_count 值;
        }
      While(该存储块为空闲并且写次数小于
        K×avg_count);
      将原存储块中数据复制到新的存储块
        中;
      修改对应文件的存储块链表;
      从离线迁移队列中删除该地址;
    }
    Else
      休眠 Sleep(T);
  }
}
```

热数据主动迁移策略中,无需扫描整个存储空间,在写数据之前主动监测,发现哪些存储块存在写次数过多的问题;为了不影响文件系统的 I/O 性能,采用离线的数据迁移线程,在 I/O 负载较低的情况下进行数据的迁移;在迁移时与文件分配类似,选择写次数较低的存储块作为迁移的目标存储块,避免频繁的数据迁移.

3 测试与分析

我们实现了面向新型存储器的文件系统级磨损均衡机制的原型,使用通用测试工具和多种数据集进行测试与分析,检验面向新型存储器的文件系统级磨损均衡机制的效果.

3.1 原型系统与测试环境

为了有效和方便地测试面向新型存储器文件系统级磨损均衡机制的效果和造成的额外开销,我们在开源基于对象存储设备 Open-osd 上,修改 OSD Target 模块中的代码,实现面向新型存储器的文件系统级磨损均衡机制的原型. 首先使用 DRAM 模拟新型储器构建一定容量的外存系统,修改 Open-osd 系统创建文件的过程、加入文件均匀分布策略,在 Open-osd 系统的写操作流程中加入热数据主动迁移策略,并依据文件名采用 Hash 算法查找存储块,最后增加了记录各个存储块写次数模块和记录热数据置换次数模块,便于分析文件系统级磨损均衡机制的磨损均衡效果. 同时使用 RAMFS 代替磁盘作为 Open-osd 的存储介质,与面向新型存储器文件系统级磨损均衡机制原型进行对比测试. 测试时固定设置 $K=0.7$, $S=3$,保持磨损均衡效果和对 I/O 性能影响的均衡.

为了减少测试时网络通信对 I/O 性能的影响,使用一台机器搭建测试环境,配有 Intel® Xeon® E5606 2.13 GHz 处理器和 16 GB 的内存. 使用 postmark 和 filebench 作为测试工具,测试面向新型存储器文件系统级磨损均衡机制原型的磨损均衡效果和 I/O 性能,与运行于 RAMFS 上的 Open-osd 进行对比测试与分析. postmark 是一款通用的文件系统读写测试工具,用于模拟真实应用程序服务器的负载;同时使用 filebench 测试将文件系统用于支持 email 或电子商务等更新数据频繁的应用时的磨损均衡效果和额外开销.

3.2 磨损均衡效果测试

所有存储块的最大写次数和最小写次数之差是磨损均衡效果的一个很好参考标准,我们使用 postmark 和 filebench 测试工具测试使用文件系统级磨损均衡机制前后所有存储块的最大写次数和最小写次数,计算二者的差值.

首先使用 filebench 和 postmark 测试使用面向新型存储器的文件系统级磨损均衡机制前后所有的

存储块最大和最小写次数,由此计算出最大和最小写次数的差值,使用 filebench 时选择需要频繁修改文件的 varmail 和 fileserver 负载,设置文件的大小分别为 1～9 KB,10～63 KB,64～1 023 KB 和 1 024～4 096 KB,测试测试结果如图 1～3 所示:

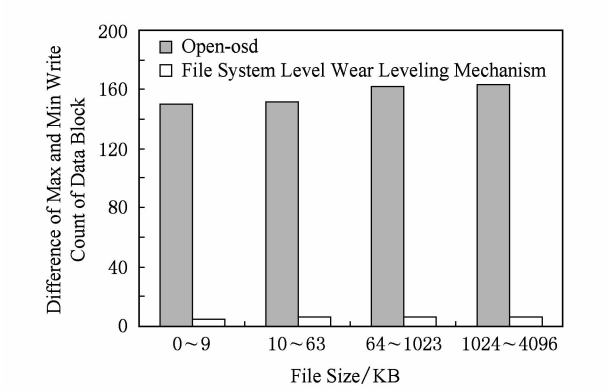

Fig. 1　Difference of write_count under workload of varmail.

图 1　varmail 数据集测试最大和最小写次数差

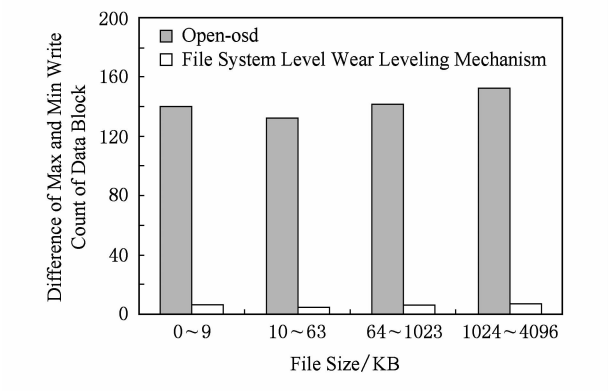

Fig. 2　Difference of write_count under workload of fileserver.

图 2　fileserver 数据集测试最大和最小写次数差

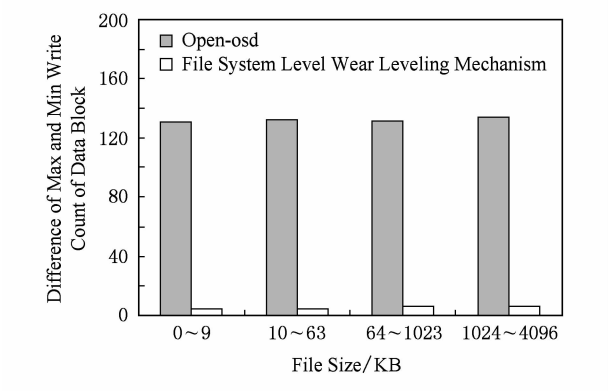

Fig. 3　Difference of write_count under workload of postmark.

图 3　postmark 数据集测试最大和最小写次数差

从图 1～3 中可以发现,面向新型存储器的文件系统级磨损均衡机制能明显降低存储块的写次数差别,降低到了原来的 1/20 左右. 这是由于文件均匀分布算法采用 Hash 函数分配文件空间,能将文件较均匀地分布到外存中,在分配存储块时选择写次数较低的存储块避免了频繁分配同一存储块造成的写操作集中问题;同时热数据主动迁移算法会迁移写次数较高的存储块,分散写操作. 此外从图 1～3 中可以发现在负载和文件大小不同时,面向新型存储器的文件系统级磨损均衡机制性能稳定,均达到了良好的磨损均衡效果.

3.3　I/O 性能测试

面向新型存储器的文件系统级磨损均衡机制会增加一定的写操作,从而对 Open-osd 的 I/O 性能产生一定的影响,我们使用 filebench 的 varmail,randomrw 数据集和 postmark 测试使用面向新型存储器的文件系统级磨损均衡机制后系统的 I/O 性能,与运行于 RAMFS 上的 Open-osd 系统进行对比,测试结果如图 4～6 所示:

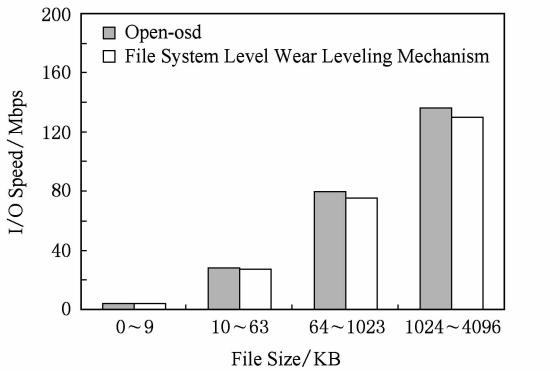

Fig. 4　Performance of I/O under workload of varmail.

图 4　varmail 数据集测试 I/O 性能

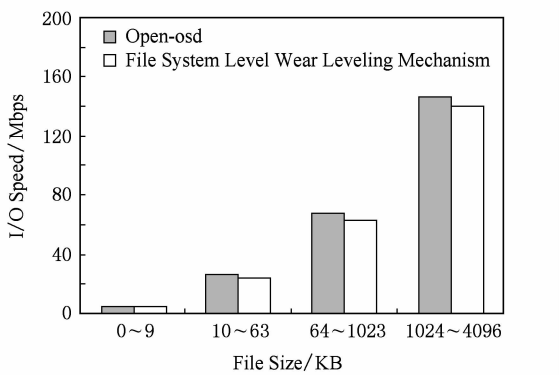

Fig. 5　Performance of I/O under workload of randomrw.

图 5　randomrw 数据集测试 I/O 性能

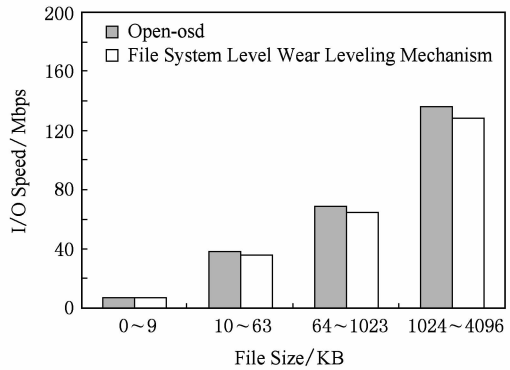

Fig. 6 Performance of I/O of postmark.

图 6 postmark 数据集测试 I/O 性能

从图 4～6 中可以发现,使用面向新型存储器的文件系统级磨损均衡机制后,系统的 I/O 性能下降很小,仅减低了 4.5%～6%,这是由于文件均匀分布策略已经很好地将文件分散到了外存系统中,同时避免新文件使用被经常修改或被删除旧文件的存储空间带来的写操作集中问题,大大减少了需要迁移的存储块数量,同时采用利用 I/O 负载较低的时段才迁移存储块的测试写次数过多的存储块,从而有效避免了磨损均衡机制对 I/O 性能的影响.

从图 4～6 中还可以发现在不同负载和不同文件大小的测试环境中,I/O 性能的下降变化幅度仅在 2%以内,这表明面向新型存储器的文件系统级磨损均衡机制能适应不同的运行环境,使得系统始终能保持较高的 I/O 性能.

3.4 写数量测试

存储块的迁移会增加额外的写操作,我们使用 filebench 下的数据集 varmail,randomrw 和 postmark,设置文件大小区间分别为 1～9 KB,10～63 KB,64～1 023 KB 和 1 024～4 096 KB,设定写数据量为 1 GB,测试面向新型存储器的文件系统级磨损均衡机制所增加的写数据量,结果如图 7～9 所示.

从图 7～9 中可以发现,使用面向新型存储器的文件系统级磨损均衡机制后,写数据量仅增加了 0.5%～0.1%. 这是由于文件均匀分布算法使用 Hash 函数把文件均匀分布到磁盘上,同时分配存储空间时避免了写次数较高的存储块,从而预防了单个存储块写次数过多的情况,减少了需要迁移的数据块数量.

从图 7～9 中还可以发现,使用不同负载和设置不同文件大小时写数据量的增加基本保持稳定,仅随着文件大小的增加下降了 0.4%;这是由于文件系统中通常元数据的修改次数远高于文件,大文件中文件数量较少,随之元数据量也较低,从而减少了

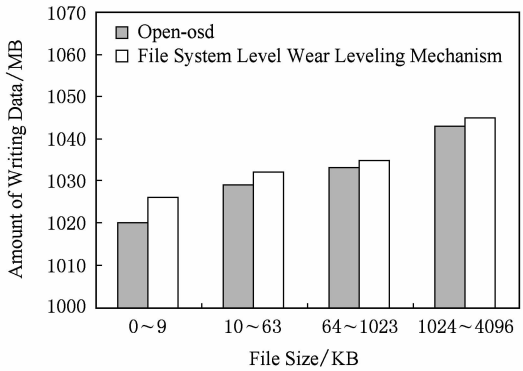

Fig. 7 Amount of writing data under workload of varmail.

图 7 varmail 数据集测试写数量

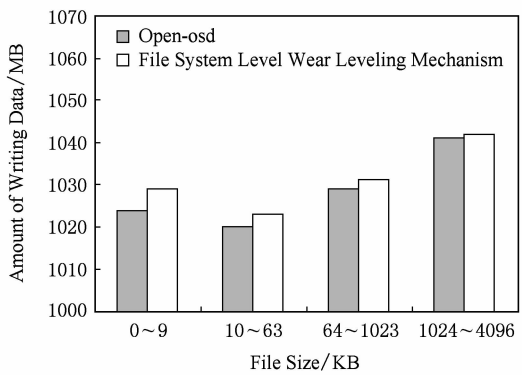

Fig. 8 Amount of writing data under workload of randomrw.

图 8 randomrw 数据集测试写数量

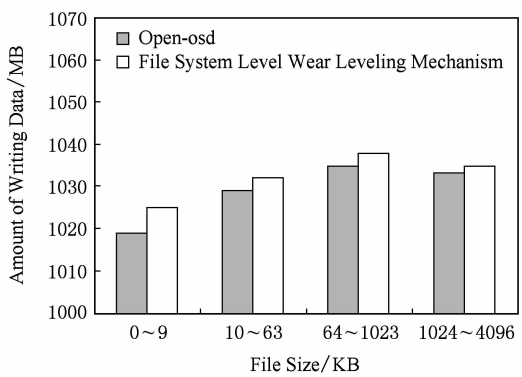

Fig. 9 Amount of writing data of postmark.

图 9 postmark 数据集测试写数量

需要迁移的存储块数量,降低了采用面向新型存储器的文件系统级磨损均衡机制后额外的写数据量.

4 总 结

高效和性能稳定的磨损均衡机制是使用新型非

易失存储器构建外存系统时急需解决的重要问题.本文设计了一种文件系统级的磨损均衡技术,使用BKDRHash函数将文件分散在外存系统中,通过选择写次数较少的存储块存储文件,避免创建和删除文件时重复使用相同存储块带来的写操作集中问题,减少需要迁移存储块的数量;使用主动迁移策略,在外存系统I/O负载较低时主动迁移写操作密集的存储块,避免对外存系统I/O性能的影响.并在开源的基于对象存储设备Open-osd上,实现了面向新型存储器文件系统级磨损均衡机制的原型,使用存储系统通用测试工具filebench和postmark,以及多个通用数据集进行了测试与分析,结果表明面向新型存储器文件系统级磨损均衡机制能稳定地将存储块的写次数差减少到原来的1/20左右,同时最高仅降低了6%的I/O性能和增加了0.5%的写数据量,从而验证了面向新型存储器的文件系统级磨损均衡机制具有高效和稳定的特性.

我们所设计的面向新型存储器文件系统级磨损均衡机制仍然使用存储块来管理外存空间,没有利用新型存储器支持字节寻址的特性,不利于存储和管理小文件.未来将针对外存空间管理的灵活性问题开展相关的研究.

参　考　文　献

[1] Caulfield A M, De A, Coburn J, et al. Moneta: A high-performance storage array architecture for next-generation, non-volatile memories [C] //Proc of the 43rd Annual IEEE/ACM Int Symp on Microarchitecture. Piscataway, NJ: IEEE, 2010: 385-395

[2] Dhiman G, Ayoub R, Rosing T. PDRAM: A hybrid PRAM and DRAM main memory system [C] //Proc of the 46th IEEE/ACM Annual Design Automation Conf. Piscataway, NJ: IEEE, 2009: 664-669

[3] Qureshi M K, Srinivasan V, Rivers J A. Scalable high performance main memory system using phase-change memory technology [J]. ACM SIGARCH Computer Architecture News, 2009, 37(3): 24-33

[4] Hu J, Zhuge Q, Xue C J, et al. Software enabled wear-leveling for hybrid PCM main memory on embedded systems [C] //Design, Automation and Test in Europe Conf and Exhibition. Piscataway, NJ: IEEE, 2013: 599-602

[5] Wu X, Reddy A L. SCMFS: A file system for storage class memory [C] //Proc of the 2011 Int Conf for Gigh Performance Computing, Networking, Storage and Analysis. Piscataway, NJ: IEEE, 2011: 1-19

[6] Condit J, Nightingale E B, Frost C, et al. Better I/O through byte-addressable, persistent memory [C] //Proc of the ACM SIGOPS 22nd Symp on Operating Systems Principles. New York: ACM, 2009: 133-146

[7] Yue J, Zhu Y. Accelerating write by exploiting PCM asymmetries [C] //Proc of the Int Conf on High Performance Computer Architecture. Piscataway, NJ: IEEE, 2013: 282-293

[8] Fang Y, Li H, Li X. Life time enhancement techniques for PCM based image buffer in multimedia applications [J]. Very Large Scale Integration Systems, 2009, 99(1): 129-136

[9] Chang Y H, Hsieh J W, Kuo T W. Improving flash wear leveling by proactively moving static data [J]. IEEE Trans on Computers, 2010, 59(1): 53-65

[10] Chen F, Luo T, Zhang X. CAFTL: A content-aware flash translation layer enhancing the lifespan of flash memory based solid state drives [C] //Proc of the 9th USENIX Conf on File and Storage Technologies. Berkeley: USENIX Association, 2011: 77-90

[11] Lee S W, Sang Wong, Moon B. Design of flash-based DBMS: An in page loging approach [C] ///Proc of the 2007 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2007: 55-66

[12] Wu G, He X. Delta-FTL: Improving SSD lifetime via exploiting content locality [C] //Proc of the 7th ACM European Conf on Computer Systems. New York: ACM, 2012: 253-266

[13] Pure Storage. How flash changes everything [OL]. [2012-09-01]. http://www.purestorage.com/pdf/Pure_storage_whitepaper_how_flash_changes_everything.pdf

[14] XtremIO. Flash implications in enterprise storage arraydesigns [OL]. [2012-09-01]. http://ww.Xtremio.com/resources/xtremio-white-paper-flash-implications-inenterprisc-storage_array_designs

[15] Lu Youyou, Shu Jiwu. Summary of flash memory storage system [J]. Journal of Computer Research and Development, 2013, 50(1): 49-59 (in Chinese)
(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)

[16] Soundararajan G, Prabhakaran V, Balakrishnan M, et al. Extending SSD left times with disk based write caches [C] //Proc of the 8th USENIX Conf on File and Storage Technologies. Berkeley: USENIX Association, 2010: 101-114

[17] Liu D, Wang T, Wang Y, et al. Curling-PCM: Application-specific wear leveling for phase change memory based embedded systems [C] //Proc of Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2013: 279-284

Cai Tao, born in 1976. Received his PhD degree in computer science in 2008. Associate professor of Jiangsu University. Member of China Computer Federation and ACM. His current research interests include network storage, cloud storage and non-volatile memory.

Zhang Yongchun, born in 1988. Master. His main research interests include information storage and non-volatile memory.

Ni Xiaorong, born in 1989. Master. Her main research interests include information storage and NVM.

Niu Dejiao, born in 1978. PhD candidate. Associate professor. Her current research interests include file system, cloud storage and network storage.

Liang Dongying, born in 1976. Master, senior engineer. His main research field is about embedded system.

2015 年《计算机研究与发展》专题(正刊)征文通知

——面向“互联网+”的应用技术

“互联网+”通过发挥互联网、计算机技术在生产要素配置中的优化和集成作用,提升实体经济的创新力和生产力. 以 MOOC 为代表的互联网教育、以工业 4.0 为代表的智能制造、以互联网金融为代表的现代服务业等,都展示出“互联网+”的巨大潜力.“互联网+”是计算机与其他学科交叉而成的新型技术,面临一系列理论和技术的挑战,如物理系统的状态感知、多维异构数据的融合分析、服务模式的优化改进、多目标的联合决策等,是一项极富有挑战性的研究内容.

为推动我国学者在“互联网+”领域的研究、及时报道我国学者的最新研究成果,《计算机研究与发展》将于 2015 年 12 月出版应用技术专辑——面向“互联网+”的应用技术,欢迎相关领域的专家学者和科研人员踊跃投稿.

征文内容(包括但不限于)

—大数据分析挖掘技术的典型应用
- 面向典型应用的大数据并行计算模式
- 面向典型应用的大数据可视化
- 面向典型应用的大数据真实性检测方法
- Web 大数据挖掘技术典型应用
- 行业大数据分析技术典型应用
- 企业大数据挖掘技术典型应用

—大规模移动学习
- 大规模泛在移动学习服务模式研究
- 大规模泛在移动学习的云、网、端支撑环境与体系架构研究
- 移动/无线网络多媒体传输技术研究
- 大规模移动学习日志的存储、分析与应用技术研究
- 移动学习个性化资源推荐技术研究
- 大规模移动学习的过程管理与评价方法研究
- 图像、音频和视频的建模与理解算法
- 多媒体标注、检索和推荐应用
- 多媒体安全与内容保护

—信息物理融合系统
- 信息物理融合系统建模方法
- 信息物理融合系统的服务模式和新兴应用
- 现代制造系统的优化控制
- 智能楼宇和智能家庭
- 智能医疗
- 智能交通
- 智能电网

投稿要求

1) 论文应属于作者的科研成果,数据真实可靠,具有重要的学术价值与推广应用价值,未在国内外公开发行的刊物或会议上发表或宣读过,不存在一稿多投问题. 作者在投稿时,需向编辑部提交版权转让协议.

2) 论文应包括题目、作者信息、摘要、关键词、正文和参考文献,论文一律用 word 排版,论文格式体例格式请参考《计算机研究与发展》近期文章.

3) 论文需附通讯作者的联系地址、电话或手机及 E-mail 地址.

4) 论文请通过期刊网站(http://crad.ict.ac.cn)进行投稿,并在作者留言中注明“应用技术 2015 专题”(否则按自由来搞处理).

重要日期

征文截止日期: 2015 年 8 月 10 日　　录用通知日期:2015 年 9 月 20 日

作者修改稿提交日期:2015 年 10 月 1 日　　出版日期:2015 年 12 月

特邀编委

郑庆华　教授　西安交通大学　qhzheng@mail.xjtu.edu.cn

联系方式

联系人:郑庆华　qhzheng@mail.xjtu.edu.cn

编辑部　crad@ict.ac.cn　010-62620696, 010-62600350

通信地址: 北京 2704 信箱《计算机研究与发展》编辑部　邮政编码: 100190