

## 应用透明的超算多层存储加速技术研究

何晓斌<sup>1</sup>,高洁<sup>1</sup>,肖伟<sup>1</sup>,陈起<sup>2</sup>,刘鑫<sup>1</sup>,陈左宁<sup>3</sup>

(1.国家并行计算机工程技术研究中心,北京 100080; 2.清华大学 计算机科学与技术系,北京 100084;

3.中国工程院,北京 100088)

**摘要:**在E级计算时代,超算系统一般使用多层存储架构以满足应用数据访问的容量和性能需求,这种架构中不同层次的存储介质差异较大,难以实现统一名字空间管理,往往需要应用修改数据访问流程才能最大程度利用到多层存储的性能和容量优势。针对多层存储统一名字空间的问题,提出针对非易失性双列存储模块(NVDIMM)的块级缓存和针对突发缓冲存储(BB)的文件级缓存技术。基于NVDIMM的块级缓存技术对缓存窗口灵活控制,以支持数据块粒度的异步读写,实现NVDIMM与BB层统一名字空间管理;基于BB的文件级缓存技术将数据缓存在BB层中,并动态迁移和管理文件副本,实现BB层与传统磁盘文件系统统一名字空间管理。在神威E级原型验证系统中的测试结果表明,所提出的两种技术较好地解决了多层存储的透明加速难题,NVDIMM块级缓存与BB相比,在缓存窗口16 MB时128 KB顺序读写带宽分别提升27%和36%,8 KB随机读写带宽分别提升20%和37%;基于BB的文件缓存技术利用BB的高带宽支撑数据访问,与全局文件系统相比,128 KB顺序读写带宽分别提升55%和141%,8 KB随机读写带宽分别提升163%和209%。此外,实际应用的测试也表明以上两种缓存技术具有透明的存储加速效果。

**关键词:**超算系统;分层存储;非易失性双列存储模块;突发缓冲存储;块级缓存;文件级缓存;透明加速

开放科学(资源服务)标志码(OSID):



中文引用格式:何晓斌,高洁,肖伟,等.应用透明的超算多层存储加速技术研究[J].计算机工程,2022,48(12):1-8.

英文引用格式:HE X B,GAO J,XIAO W,et al.Research on application-transparent supercomputing multi-tier storage acceleration technology[J].Computer Engineering,2022,48(12):1-8.

## Research on Application-Transparent Supercomputing Multi-tier Storage Acceleration Technology

HE Xiaobin<sup>1</sup>,GAO Jie<sup>1</sup>,XIAO Wei<sup>1</sup>,CHEN Qi<sup>2</sup>,LIU Xin<sup>1</sup>,CHEN Zuoning<sup>3</sup>

(1.National Research Center of Parallel Computer Engineering and Technology,Beijing 100080,China; 2.Department of Computer Science and Technology,Tsinghua University,Beijing 100084,China; 3.Chinese Academy of Engineering,Beijing 100088,China)

**[Abstract]** Supercomputing systems have entered the exascale era and to meet the growing demand for data access and performance, supercomputing systems typically require multi-tier storage architecture. Storage media typically vary between tiers, making it difficult to achieve unified namespace management. This paper proposes to address this problem by developing block-level caching based on Non-Volatile Dual In-line Memory Module (NVDIMM) and file-level cache based on Burst Buffer (BB) technologies. The NVDIMM-based block-level cache supports asynchronous read/write of data block granularity through flexible control of the cache window and realizes the unified namespace of NVDIMM and BB. The file-level cache is based on BB and supports data caching in it, dynamically migrates and manages file copies, and realizes the unified namespace between BB and the traditional disk file system. Evaluation result of the verification systems in the Sunway E-class prototype shows that both technologies proposed in this paper can achieve transparent acceleration. Compared with BB, the NVDIMM block-level cache increases the sequential read/write bandwidth of a 128 KB block by 27% and 36%, respectively, and the random read/write bandwidth of an 8 KB block by 20% and 37%, respectively, given a cache window size of 16 MB. The BB-based file cache allows applications to exploit the high bandwidth of BB. Compared with the Global File System (GFS), the 128 KB block sequential read/write bandwidth increases by 55% and 141%, respectively, and the 8 KB random read/write bandwidth increases by 163% and 209%, respectively. In addition, the practical application presented in this study shows that above two cache technologies have transparent storage acceleration effects.

基金项目:国家部委基金。

作者简介:何晓斌(1984—),男,副研究员、博士研究生,主研方向为分布式数据存储系统;高洁,助理研究员、硕士;肖伟,研究实习员;陈起,助理研究员、博士研究生;刘鑫,研究员、博士;陈左宁,中国工程院院士、研究员。

收稿日期:2022-10-08 修回日期:2022-12-01 E-mail:hexiaobin\_1984@163.com

**[Key words]** supercomputing system; tiered storage; Non-Volatile Dual In-line Memory Module (NVDIMM); Burst Buffer (BB); block-level cache; file-level cache; transparent acceleration

**DOI:**10.19678/j.issn.1000-3428.0065928

## 0 概述

当前高性能计算已经迈入E级时代,随着传统超算与AI的不断融合,超算应用领域不断扩大,应用产生的数据量也呈爆发性增长<sup>[1-3]</sup>。为此,主流高性能计算系统均构建了多层存储体系,主要包括全局文件系统(Global File System, GFS)、突发缓冲存储(Burst Buffer, BB)<sup>[4]</sup>、节点高速存储层等,其中GFS主要使用大容量磁盘构建,BB一般使用NVMe SSD构建。此外,近年来利用非易失性随机访问存储器(Non-Volatile Random Access Memory, NVRAM)、非易失性双列存储模块(Non-Volatile Dual In-line Memory Module, NVDIMM)等高速存储介质构建节点高速存储层也成为研究热点<sup>[5]</sup>。最新发布的美国Frontier超算部署了三层存储<sup>[6]</sup>,包括计算节点本地高速存储、全局共享的SSD存储和全局共享的磁盘存储,合计存储容量高达760 PB,数据读带宽达75 TB/s,数据写带宽达35 TB/s;日本超级计算机富岳<sup>[7]</sup>部署了两层存储,包括计算节点共享的SSD存储系统,容量达16 PB,带宽达10 TB/s,此外还基于磁盘构建了容量达100 PB的全局文件系统。存储层次的不断增加,虽然提升了存储系统的总体性能,但是目前多层存储系统的构建缺乏统一标准,特别是BB层与计算节点本地存储层,为了发挥硬件介质的性能优势,其软件设计与传统存储软件存在显著差异,无法与传统文件系统形成统一名字空间,如GekkoFS<sup>[8]</sup>、UnifyFS<sup>[9]</sup>、CROSS<sup>[10]</sup>、Datawarp<sup>[11]</sup>、BeeOND<sup>[12-13]</sup>等文件系统均基于BB构建了独立的文件系统空间,需要应用修改数据访问路径,往往会导致额外的数据管理负担。正是由于以上原因,导致存储加速系统的利用率普遍不高<sup>[14]</sup>。业界针对以上问题研发了多款软件,如Elevator<sup>[15]</sup>、Unistor<sup>[16]</sup>、Hermes<sup>[17]</sup>等,但是此类软件或者针对特定的上层I/O库(例如HDF5等),或者无法支持多层异构存储介质,难以适用于下一代国产超算存储系统。

为解决多层存储架构无法实现统一名字空间访问的问题,本文提出对应用透明的加速技术。针对计算节点NVDIMM等高速存储层设计透明的块级数据读写缓存机制,提出支持数据顺序读和跨步读的预读算法,实现NVDIMM层与BB层统一名字空间管理;针对BB层设计文件级的数据缓存机制,提出缓存副本的一致性管理方法,实现BB层与传统GFS层统一名字空间管理。最终,基于神威E级原型验证系统对这2种缓存技术进行测试。

## 1 相关工作

当前,随着超算算力的高速增长,为支撑超算应用的数据访问需求,超算存储系统一般使用分层存

储架构构建,主要包括大容量全局文件系统、高性能Burst Buffer以及计算节点本地高速NVDIMM等层次<sup>[6-7,18-19]</sup>,其架构如图1所示。

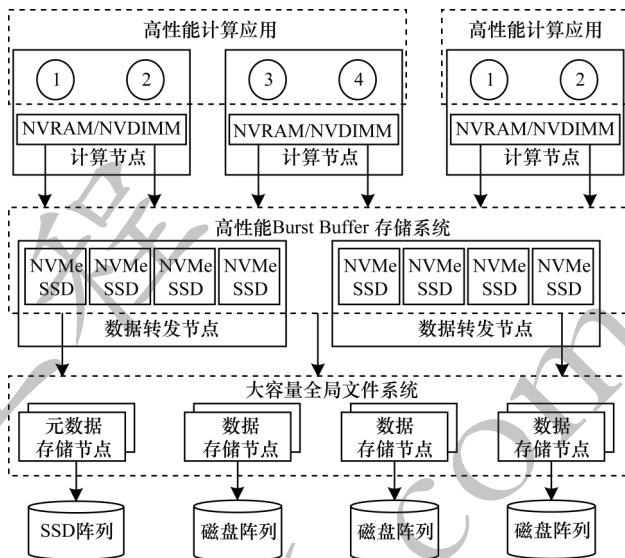


图1 超算分层存储架构

Fig.1 Tiered storage architecture of supercomputing

大容量全局文件系统一般基于磁盘存储构建,具有容量大、数据全局共享、可靠性高、兼容标准接口、聚合带宽较低等特点,如Lustre<sup>[20-21]</sup>、BeeGFS<sup>[12]</sup>等;而高性能BB存储系统一般基于SSD构建,性能相比全局文件系统更高,具备一定的数据共享机制,但是目前这一类系统往往不能完整支持POSIX接口,应用使用时不仅需要修改路径,而且还需要定制数据从GFS导入至BB(stage-in)以及从BB导出至GFS(stage-out)的管理流程,如GekkoFS<sup>[8]</sup>、DAOS<sup>[22]</sup>、UnifyFS<sup>[9]</sup>、HadaFS<sup>[23]</sup>、Datawarp<sup>[11]</sup>等;计算节点本地的NVDIMM存储性能更接近内存,但是容量较小,仅可作为临时性局部存储空间,这类存储介质一般通过块接口使用<sup>[24-25]</sup>,难以适用于通用的I/O流程<sup>[26]</sup>。

分层存储通过更多层次高速硬件的部署提升了存储系统的基础性能,然而却带来了统一名字空间的问题,如何对应用隐藏多层次存储系统的复杂性,透明提升应用数据访问的效率成为业界关注的难题<sup>[14,27]</sup>。Elevator提出了一种针对Burst Buffer的透明缓存机制,可实现BB与GFS的透明整合,但是这种机制只能运行在HDF5库内部,存在适应性问题<sup>[15]</sup>;Unistor在Elevator的基础上实现了对通用场景下BB的加速机制支持,但是无法支持NVDIMM存储环境<sup>[16]</sup>;IBM在Summit中分别提出了面向计算节点本地SSD的块和文件级缓存机制,但是仅支持透明的写缓存<sup>[28]</sup>;LPCC基于Lustre文件系统实现了BB与GFS的数据缓存<sup>[20]</sup>,但是其要求BB中的数据必须写回GFS以支持数据共享,数据共享效率较低;

Hermes提出多级存储融合使用的方法,但是其缓存管理和分配机制需要统计大量I/O事件信息,在大规模超算中存在扩展性差的问题<sup>[17]</sup>。

神威E级原型验证系统是我国面向E级超算关键技术挑战构建的验证平台<sup>[29]</sup>,配置了全局文件系统、BB存储系统,并且具备利用节点内存进行I/O加速的基础。该系统中的全局文件系统基于Lustre+LWFS构建,与神威·太湖之光全局存储相似<sup>[19]</sup>;BB存储系统基于全新研制的HadaFS构建,该系统实现了用户层、松耦合的文件访问语义。面向下一代超算的I/O挑战,本文针对NVDIMM的块访问和BB的文件访问优势分别提出不同的数据缓存机制,从而支持多层存储加速的统一名字空间,实现对应用透明的I/O加速。相关技术在神威E级原型验证系统中进行部署,为神威下一代超算存储系统的研发提供参考。

## 2 透明存储加速技术的设计

为解决当前超算多层存储异构软件堆叠带来的多种名字空间统一的问题,本文提出面向下一代超算的透明存储加速技术,如图2所示,该技术嵌入在超算多层存储软件栈中,支持POSIX标准的文件读写接口,主要包括基于NVDIMM的块级缓存和基于BB的文件级缓存两个方面。基于NVDIMM的块级缓存以计算节点高速NVDIMM存储介质为基础,通过NVDIMM高速块访问协议实现对传统文件数据块的缓存;基于BB的文件级缓存通过文件级的动态数据迁移,解决BB层与全局文件系统无法实现名字空间统一的问题。

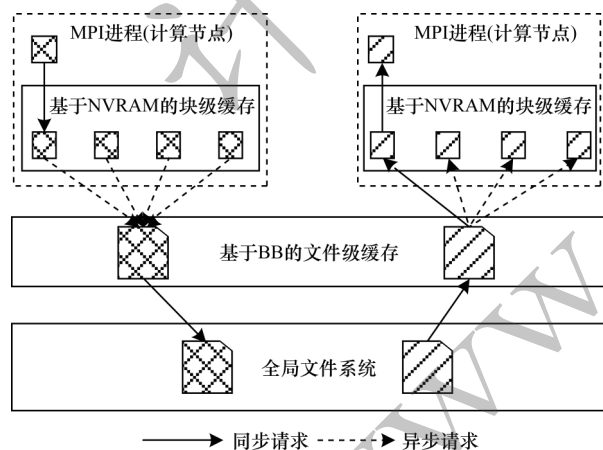


图2 面向下一代超算的透明存储加速技术架构

Fig.2 Transparent storage acceleration technology architecture for next-generation supercomputing

### 2.1 基于NVDIMM的块级缓存

基于NVDIMM的块级缓存运行在超算BB层存储之上,针对计算节点本地读写的数据流进行加速,其利用计算节点NVDIMM实现对文件的数据块级缓存,原理如图3所示。NVDIMM层块级缓存功能作为一个独立模块嵌入超算存储客户端内部,自动截获应用文件读写的数据块地址、大小等信息,并为存储客户端提供块级写数据缓存和数据预读功能。

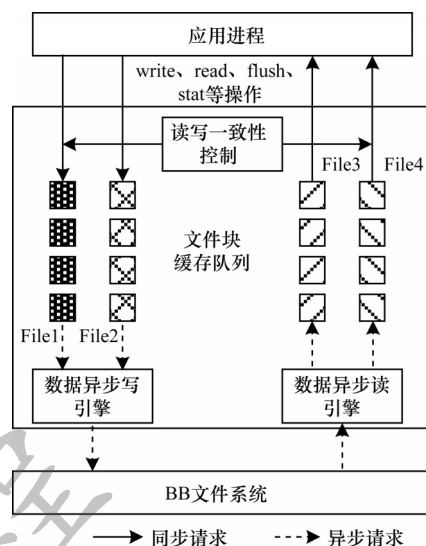


图3 NVDIMM块级缓存的工作原理

Fig.3 Principle of NVDIMM block-level cache

NVDIMM块级缓存模块在NVDIMM介质中以文件为单位分别维护了读写两个先进先出(FIFO)队列,此外还包括数据一致性控制、数据异步写引擎和数据预读引擎等部分。应用进程产生的文件操作如读(read)、写(write)、刷新(flush)、获取状态(stat)等将按照时间顺序依次进入队列,并由数据一致性控制部件决定是否立即向上层应用返回成功。对于写操作,若NVDIMM中事先设定的缓存窗口没有溢出,则设置为异步操作,否则设置为同步操作。异步操作意味着该写请求的数据块将被缓存至NVDIMM,同时客户端将立即向应用返回写成功,而同步操作意味着客户端需要等待该请求刷新至BB层文件系统后才能向应用返回结果。写操作缓存的数据块由数据异步写引擎负责刷新至BB层文件系统中。

对于读操作,系统将尽可能多地预读数据块并缓存在读数据队列中,以减少从服务端直接读数据的次数。数据读引擎根据读缓存队列中缓存的数据块大小和偏移位置决定预读数据起始位置和大小,目前有可探测顺序(Sequential)读和跨步(Stride)读两种模式,对于随机读默认按照顺序读的方式预取数据,数据预取块大小为事先设定的大数据块,以提升命中概率。数据预读算法如算法1所示。

#### 算法1 数据预读算法

输入 最近两次读请求  $q1(offset, size)$ 、 $q2(offset, size)$ , 以及记录的跨步偏移量  $stride\_offset$   
 输出 预读的数据块请求  $q\_preread(offset, size)$   
 1. if ( $q1.offset + q1.size == q2.offset$ ) //Sequential read  
 2.  $q\_preread = (q2.offset + q2.size, BLOCK\_SIZE)$   
 3. elif ( $stride\_offset \neq 0$ )  
 4. if ( $stride\_offset == q2.offset - q1.offset$ ) //Stride read  
 5.  $q\_preread = (q2.offset + stride, q2.size)$   
 6. else //Random read  
 7.  $q\_preread = (q2.offset + q2.size, BLOCK\_SIZE);$   
 8. else //Suppose to be stride read  
 9.  $stride\_offset = q2.offset - q1.offset$





### 3 测试与结果分析

#### 3.1 测试环境

神威系列超算是具有世界影响力的超算平台<sup>[30-31]</sup>,神威E级原型验证系统是神威面向E级时代超算技术研制的验证系统,部署于国家超级计算济南中心,其基于国产SW26010Pro处理器构建<sup>[32]</sup>,每个处理器包含6个核组,可支持6路I/O并发进程,节点之间使用神威网络互连<sup>[29]</sup>,单节点上网带宽达到400 Gb/s以上,可用于I/O的网络带宽超过200 Gb/s。神威E级原型验证系统的存储系统符合图1所示的超算存储架构,其中数据转发节点配置了双路服务器CPU,单节点网络带宽为200 Gb/s,并部署了两块NVMe SSD(单块容量为3.2 TB)以构建突发缓冲存储(BB)。全局存储方面利用Lustre构建了全局文件系统(GFS),利用LWFS实现了全局文件系统的数据转发。在本文的测试中,由于NVDIMM硬件针对国产处理器环境的适配仍在实验阶段,因此使用节点DRAM内存模拟NVDIMM构建块级缓存。

#### 3.2 测试方法

本文使用IOR Benchmark<sup>[33]</sup>在神威原型验证系统中测试存储加速技术使用后的带宽变化情况。IOR测试进程运行在计算节点中,每个测试进程通过本文提出的加速技术读写独立的文件。测试在不修改应用程序数据访问路径的前提下进行,分别测试加速技术启用和不启用时的性能,以对比本文所提技术的透明加速效果。此外,还使用超算典型应用对本文提出的加速技术进行测试。

#### 3.3 测试结果分析

##### 3.3.1 计算节点NVDIMM块级缓存的测试

NVDIMM块缓存机制运行在存储系统客户端,由于在高性能应用运行时,NVDIMM可能有SWAP空间、应用直接存储数据等多种其他用途,同时为了测试NVDIMM缓存机制在缓存溢出时的加速效果,选择16 MB缓存窗口进行测试(单节点4进程运行时占据的内存为64 MB,考虑常用的国产NVDIMM单根大小为16 GB,占比为1/256),对比技术为未使用NVDIMM块缓存的BB文件系统HadaFS的常规读写机制,内存刷新和预读块大小为1 MB,选择128 KB块顺序读写和8 KB块随机读写两种典型模式,128 KB块顺序读写单进程数据量为1 GB,8 KB块随机读写单进程数据量为80 MB,由于NVDIMM块缓存机制仅作用在客户端,因此测试中服务端固定为一个转发节点。

缓存窗口大小为16 MB时128 KB顺序读写带宽的测试结果如图6所示。随着测试进程数的不断增加,块缓存机制实现了读写带宽的加速。对于写操作:当1、4、16进程时,块缓存模式写带宽增幅超过40%;当进程超过64个时,写带宽增幅逐渐降低;当1 024个进程并发时,增幅为12%;不同进程规模下的平均增幅为36%。出现这种现象的主要原因在于块缓存机制启用后虽然数据写入本地缓存,但是由于缓存窗口大小的限制,并未实现完全的缓存命中。以64进程为例,写

缓存命中率为89%,一旦写缓存未命中,则该请求必须等待缓存队列清空后才能向应用返回,因此影响了性能。此外,测试中计算了文件的close时间,由于写文件完成后的close操作需要刷新元数据等信息,因此耗时较长,也一定程度影响了写缓存的总体效果。对于读操作:由于数据预读的块大小为1 MB,128 KB块读写时缓存命中率最高为87.5%,命中率低于写操作,因此测试的不同进程规模下的平均增幅更小,为27%;随着进程数的不断增加,加速幅度降低的主要原因在于底层存储的性能已经达到峰值,因此每个计算节点端可见的缓存刷新或者预读的速率降低,从而导致缓存命中率降低。

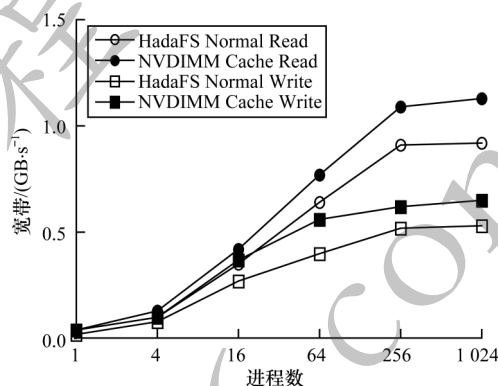


图6 16 MB缓存窗口时128 KB块的顺序读写性能

Fig.6 Sequential read/write performance of a 128 KB block when the cache window is set to 16 MB

缓存窗口大小为16 MB时8 KB随机读写带宽如图7所示。随着测试进程数的增加,8 KB随机读写的带宽不断增加,相对HadaFS常规操作的性能增幅变化与128 KB块大小顺序读写带宽增幅变化规律相似,在64进程规模下读写带宽分别提升19%和39%,不同进程规模下的平均增幅分别为20%和37%。虽然8 KB随机场景下NVDIMM缓存机制难以准确预测出读请求的偏移量,但是由于缓存窗口相对于8 KB块大小比例较高,因此提前读入的数据量大,一定程度上增加了缓存命中的概率,由此保证了读带宽相比未使用缓存机制有所提升。

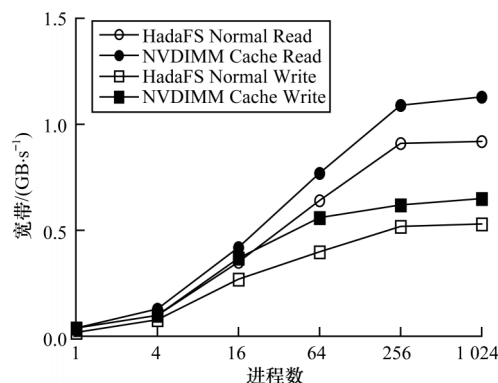


图7 16 MB缓存窗口8 KB块的随机读写性能

Fig.7 Random read/write performance of an 8 KB block when the cache window is set to 16 MB

固定进程数为1 024,数据块大小为128 KB,每个进程顺序读写1 GB数据,并按照4倍比例逐渐增加缓存窗口和数据预读块,测试得到的带宽数据归一化后如图8所示。随着缓存窗口的不断增大,写缓存命中率不断提升,因此写带宽加速比也不断增加,一旦缓存窗口大小超过写入的数据总量,写缓存的命中率达到100%,写带宽达到峰值,1 GB窗口下写带宽相比16 MB提升达到10倍以上。随着缓存窗口、预读数据块大小的不断增加,读带宽也呈现增加趋势,但是增加幅度相对缓慢,这是由于预读数据块大小始终没有超过文件大小。固定测试的进程规模,单进程以8 KB块随机读写80 MB数据,在增加缓存窗口时呈现出完全相同的规律,且加速效果更加明显,此处不再赘述。

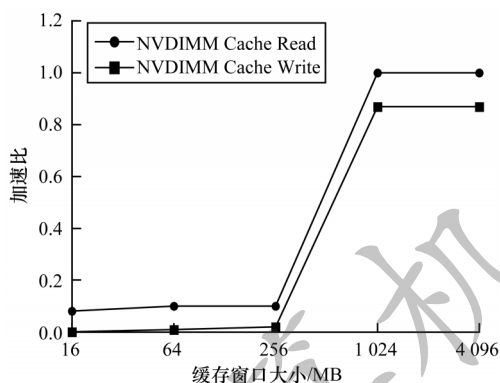


图8 增加缓存窗口时128 KB块的顺序读写性能

Fig.8 Sequential read/write performance of a 128 KB block when increasing cache window size

以上测试表明,未来随着NVDIMM等新型存储介质的规模应用,对于文件大小在NVDIMM缓存空间内的应用,块缓存机制有望大幅提升应用的数据读写性能,对于那些文件大小超过缓存空间的应用,该机制也具有一定的加速效果。

### 3.3.2 基于BB的文件级缓存测试

基于Burst Buffer的文件级缓存机制运行在BB节点上,负责实现BB存储的HadaFS文件系统与底层全局文件系统(GFS)之间的数据统一管理和透明加速。测试使用IOR程序,分别测试128 KB顺序读写和8 KB随机读写两种模式,对比技术为使用磁盘构建的GFS。测试中关闭了计算节点NVDIMM缓存,测试程序运行在计算节点,分别测试1、16、256、1 024、4 096、8 192进程规模,128 KB块顺序读写测试每个进程读写数据量为1 GB,8 KB块随机读写测试每个进程读写数据量为80 MB,服务端运行在8个转发节点,每个转发节点包含2个NVMe SSD,对比本文提出的BB级文件缓存功能打开和关闭(相当于使用底层GFS进行数据读写)时的聚合带宽变化,以测试加速效果。

128 KB块顺序读写性能如图9所示。随着进程规模的不断增加,GFS模式和BB文件级缓存模式均实现了性能增长,最终在8 192进程规模下BB文件级缓存

模式的读写带宽相比GFS提升90%和147%,不同测试规模读写带宽的平均增幅分别为55%和141%。BB文件级缓存机制使用后,由于BB文件级缓存空间达到51.2 TB,远大于测试程序产生的数据量,确保了测试中读写数据均实现缓存命中,测试程序产生的文件全部通过SSD读写,因此,测试程序实际使用的带宽即为BB层HadaFS的峰值带宽。事实上,当前几乎所有主流超算在设计时,其BB层缓存空间基本都能满足应用突发数据的多次写入,可基本保证突发数据的缓存能通过BB层的要求,提升系统总体I/O性能。

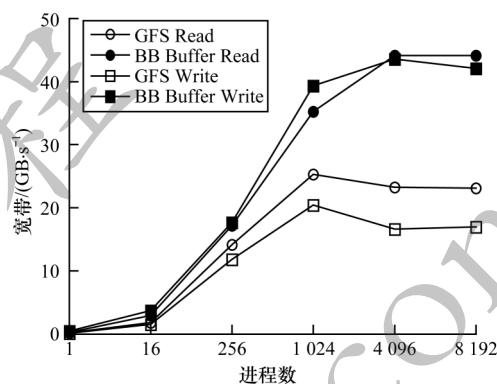


图9 BB文件级缓存128 KB块顺序读写性能

Fig.9 Sequential read/write performance of a 128 KB block in BB layer file-level cache

文件级缓存8 KB块随机读写性能如图10所示,随着进程数的不断增加,8 KB块随机读写带宽的变化规律与128 KB顺序写相似。8 192进程时8 KB随机写带宽提升240%,随机读带宽提升220%,不同测试规模下读写带宽的平均增幅分别为163%和209%,主要原因在于BB基于SSD构建,其小块随机读写性能相对于基于磁盘构建的GFS更具优势。

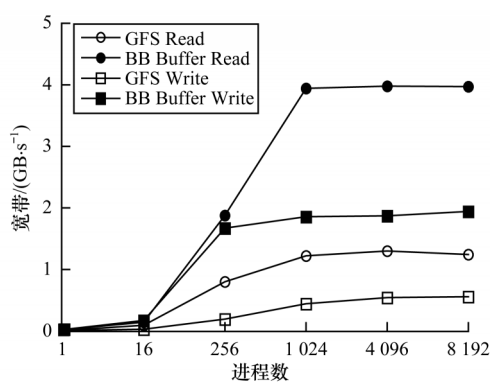


图10 BB文件级缓存8 KB随机读写性能

Fig.10 Random read/write performance of an 8 KB block in BB layer file-level cache

### 3.3.3 实际应用测试

本文提出的透明存储加速机制在神威E级原型验证系统部署后,为多道应用课题提供了数据加速服务,典型的包括WRF<sup>[34]</sup>、H5bench<sup>[35]</sup>等。

WRF是一种经典的区域数字天气预报系统,通



过单进程写出数据。使用本文提出的块级数据缓存技术后,WRF写数据I/O带宽提升50%。H5bench是一种多个进程写同一个文件的经典程序,模拟了VPIC应用的I/O流程,需要进行大量数据的读写。使用基于BB的文件级缓存后,H5bench性能变化如图11所示。可以看出:使用BB文件级缓存后,H5bench的写性能相比GFS平均提升3.6倍,读性能提升1.3倍,这是由于H5bench是典型的多进程读写同一文件的应用场景,GFS在这种场景下由于需要维护严格数据一致性,因此性能开销较大;此外,BB文件缓存的H5bench最高写带宽为图9中BB峰值性能的一半,而读性能达到图9中BB所能实现的最高读带宽,原因在于H5bench是多个进程写同一个文件,在底层BB端也面临着多个进程写同一个文件的场景,因此性能不如每个进程写独立的文件,由于多个进程读同一个文件时NVMe SSD底层文件系统缓存作用明显,因此能实现较高读性能。

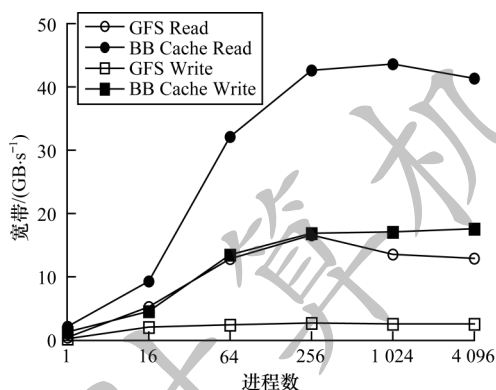


图11 H5bench的读写性能

Fig.11 Read/write performance of H5bench

#### 4 结束语

E级时代超算利用不同的存储介质构建了大容量、高性能的多层存储系统,虽然一定程度上满足了应用对于存储容量和性能的需求,但是由于多层存储异构软件使用方式上的差异,难以实现统一名字空间管理和对应用透明的缓存机制。本文针对这一问题提出基于NVDIMM的块级缓存和基于Burst Buffer的文件级缓存机制,并在神威E级原型验证系统中进行部署和测试。测试结果表明,本文提出的技术可以实现应用透明的存储加速。下一步将继续研究多层存储面向应用的资源协调与分配机制,实现多应用共享的存储加速资源动态管理。

#### 参考文献

[1] BOITO F Z, INACIO E C, BEZ J L, et al. A checkpoint of research on parallel I/O for high-performance computing[J]. ACM Computing Surveys, 2019, 51(2): 23.

[2] BHARATHI S, CHERVENAK A, DEELMAN E, et al. Characterization of scientific workflows[C]//Proceedings of the 3rd Workshop on Workflows in Support of Large-Scale

Science. Washington D. C., USA: IEEE Press, 2008: 1-10.

- [3] YASHIRO H, TERASAKI K, KAWAI Y T, et al. A 1024-member ensemble data assimilation with 3.5-km mesh global weather simulations[C]//Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis. Washington D. C., USA: IEEE Press, 2020: 1-10.
- [4] LIU N, COPE J, CARNS P, et al. On the role of Burst Buffers in leadership-class storage systems[C]//Proceedings of IEEE Symposium on Mass Storage Systems and Technologies. Washington D. C., USA: IEEE Press, 2016: 1-11.
- [5] PATIL O, IONKOV L, LEE J, et al. Performance characterization of a DRAM-NVM hybrid memory architecture for HPC applications using intel optane DC persistent memory modules[C]//Proceedings of International Symposium on Memory Systems. New York, USA: ACM Press, 2019: 1-5.
- [6] Oak Ridge National Laboratory. Frontier[EB/OL]. (2022-05-30)[2022-09-10]. <https://www.ornl.gov/news/frontier-supercomputer-debuts-worlds-fastest-breaking-exascale-barrier>.
- [7] AKIMOTO H, OKAMOTO T, KAGAMI T, et al. File system and power management enhanced for supercomputer Fugaku[EB/OL]. (2020-11-11)[2022-09-10]. <https://www.fujitsu.com/global/about/resources/publications/technicalreview/2020-03/article05.html>.
- [8] VEF M A, MOTI N, SÜB T, et al. GekkoFS—a temporary distributed file system for HPC applications[C]//Proceedings of IEEE International Conference on Cluster Computing. Washington D. C., USA: IEEE Press, 2018: 319-324.
- [9] Unifyfs. UnifyFS: a file system for Burst Buffers[EB/OL]. [2022-09-10]. <https://unifyfs.readthedocs.io/en/latest/index.html>.
- [10] 陈曦, 朱建涛, 何晓斌. 一种面向高性能计算的分布式对象存储系统[J]. 计算机工程, 2017, 43(8): 69-73.
- CHEN X, ZHU J T, HE X B. An HPC-oriented distributed object storage system[J]. Computer Engineering, 2017, 43(8): 69-73. (in Chinese)
- [11] DAVE H, BENJAMIN L, DOUG P. Architecture and design of cray datawarp[EB/OL]. (2016-05-28)[2022-09-10]. [https://cug.org/proceedings/cug2016\\_proceedings/includes/files/pap105s2-file1.pdf](https://cug.org/proceedings/cug2016_proceedings/includes/files/pap105s2-file1.pdf).
- [12] 宋振龙, 李小芳, 李琼, 等. BeeGFS并行文件系统性能优化技术研究[J]. 计算机工程与科学, 2020, 42(10): 1765-1773.
- SONG Z L, LI X F, LI Q, et al. Improving the performance of BeeGFS parallel file system[J]. Computer Engineering & Science, 2020, 42(10): 1765-1773. (in Chinese)
- [13] Thinkparq. BeeOND; BeeGFS on demand[EB/OL]. [2022-09-10]. <https://beegfs-docs.readthedocs.io/en/latest/beeond.html>.
- [14] PATEL T, BYNA S, LOCKWOOD G K, et al. Uncovering access, reuse, and sharing characteristics of I/O-intensive files on large-scale production hpc systems[C]//Proceedings of the 18th USENIX Conference on File and Storage Technologies. Santa Clara, USA: USENIX Association, 2020: 91-101.

- [15] DONG B, BYNA S, WU K S, et al. Data elevator: low-contention data movement in hierarchical storage system [C]//Proceedings of IEEE International Conference on High Performance Computing. Washington D. C. , USA: Washington D. C. , USA: IEEE Press, 2016: 152-161.
- [16] WANG T, BYNA S, DONG B, et al. UniviStor: integrated hierarchical and distributed storage for HPC [C]//Proceedings of IEEE International Conference on Cluster Computing. Washington D. C. , USA: IEEE Press, 2018: 134-144.
- [17] KOUKAS A, DEVARAJAN H, SUN X H. I/O acceleration via multi-tiered data buffering and prefetching [J]. Journal of Computer Science and Technology, 2020, 35(1): 92-120.
- [18] YANG B, JI X, MA X S, et al. End-to-end I/O monitoring on a leading supercomputer [C]//Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation. Boston, USA: USENIX Association, 2019: 379-394.
- [19] CHEN Q, CHEN K, CHEN Z N, et al. Lessons learned from optimizing the Sunway storage system for higher application I/O performance [J]. Journal of Computer Science and Technology, 2020, 35(1): 47-60.
- [20] QIAN Y, WANG F, FENG D, et al. LPCC: hierarchical persistent client caching for Lustre [C]//Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis. New York, USA: ACM Press, 2019: 1-14.
- [21] 李春艳. 面向NVM的Lustre客户端持久性缓存研究[D]. 武汉: 华中科技大学, 2019.
- LI C Y. Research on NVM-oriented Lustre persistent cache on client [D]. Wuhan: Huazhong University of Science and Technology, 2019. (in Chinese)
- [22] LOFSTEAD J, JIMENEZ I, MALTZAHN C, et al. DAOS and friends: a proposal for an exascale storage system [C]//Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis. Washington D. C. , USA: IEEE Press, 2016: 585-596.
- [23] 薛巍, 杨斌, 刘世超, 等. Beacon+: 面向E级超级计算机的轻量级端到端 I/O 性能监控与分析诊断系统 [C]//2021 CCF 全国高性能计算学术年会大会论文集. 珠海: 中国计算机学会, 2021: 1-12.
- XUE W, YANG B, LIU S C, et al. Beacon+: a scalable lightweight end-to-end I/O performance monitoring, analysis and diagnosis system for exascale supercomputer [C]//Proceedings of 2021 CCF National Annual Conference on High Performance Computing. Zhuhai, China: CCF, 2021: 1-12. (in Chinese)
- [24] WANG W, DIESTELHORST S. Quantify the performance overheads of PMDK [C]//Proceedings of International Symposium on Memory Systems. New York, USA: ACM Press, 2018: 50-52.
- [25] 朱文俊, 徐壮, 秦家佳, 等. 基于DPDK的高速存储I/O优化方法[J]. 计算机工程, 2021, 47(7): 205-211, 217.
- ZHU W J, XU Z, QIN J J, et al. DPDK-based optimization approach for high speed storage I/O [J]. Computer Engineering, 2021, 47(7): 205-211, 217. (in Chinese)
- [26] 程振京, 汪璐, 程耀东, 等. 面向高能物理分级存储的文件访问热度预测[J]. 计算机工程, 2021, 47(2): 126-132.
- CHENG Z J, WANG L, CHENG Y D, et al. File access popularity prediction for hierarchical storage for high-energy physics [J]. Computer Engineering, 2021, 47(2): 126-132. (in Chinese)
- [27] BEZ J L, KARIMI A M, PAUL A K, et al. Access patterns and performance behaviors of multi-layer supercomputer I/O subsystems under production load [C]//Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing. New York, USA: ACM Press, 2022: 1-5.
- [28] ORAL S, VAZHKUDAI S S, WANG F, et al. End-to-end I/O portfolio for the summit supercomputing ecosystem [C]//Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis. New York, USA: ACM Press, 2019: 1-14.
- [29] 高剑刚, 卢宏生, 何王全, 等. 神威E级原型机互连网络 and 消息机制 [J]. 计算机学报, 2021, 44(1): 222-234.
- GAO J G, LU H S, HE W Q, et al. The interconnection network and message machinasim of sunway exascale prototype system [J]. Chinese Journal of Computers, 2021, 44(1): 222-234. (in Chinese)
- [30] 三台E级超算原型机系统全部完成交付[J]. 河南科技, 2018(31): 6.
- Three exascale supercomputing prototype systems have been delivered [J]. Henan Science and Technology, 2018(31): 6. (in Chinese)
- [31] FU H H, LIAO J F, YANG J Z, et al. The Sunway TaihuLight supercomputer: system and applications [J]. Science China Information Sciences, 2016, 59(7): 072001.
- [32] LIU Y A, LIU X L, LI F N, et al. Closing the “quantum supremacy” gap: achieving real-time simulation of a random quantum circuit using a new Sunway supercomputer [C]//Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis. New York, USA: ACM Press, 2021: 1-5.
- [33] IOR. HPC IO benchmark repository [EB/OL]. [2022-09-10]. <https://github.com/hpc/ior>.
- [34] SKAMAROCK W C, KLEMP J B, DUDHIA J. Prototypes for the WRF(Weather Research and Forecasting) model [EB/OL]. (2001-01-01) [2022-09-10]. [https://www.researchgate.net/publication/242446432\\_Prototypes\\_for\\_the\\_WRF\\_Weather\\_Research\\_and\\_Forecasting\\_model](https://www.researchgate.net/publication/242446432_Prototypes_for_the_WRF_Weather_Research_and_Forecasting_model).
- [35] H5bench [EB/OL]. [2022-09-10]. <https://github.com/hpc-io/h5bench>.