面向湍流大数据的高效存储与访问关键技术研究

程文迪[1]，张晓[1]，潘兆辉[2]，赵友军[2]，孙晨光[1]，单学强[1]，金雨展[1]，赵晓南[1]
1. 西北工业大学计算机学院，陕西 西安 710129；
2. 西北工业大学软件学院，陕西 西安 710129

摘要
随着测量技术和数值模拟技术的发展，数据驱动的湍流研究成为该领域的新研究方法。我国已建立了多个风洞实验室和多个超算中心来模拟湍流，这些研究积累了大量的湍流数据，但是国内没有集中的湍流数据管理平台，耗资巨大的实验和仿真数据难以实现交流和共享。湍流数据具有数据量大、维度高、精度高和多源异构等特点，其存储、访问与管理存在数据集成困难、数据访问低效和存储效率低等问题。设计了一个面向航空、航天和航海典型流动问题的湍流大数据分布式存储系统TDFS。结合湍流大数据的访问特点，在TDFS中设计了新的元数据组织方式和数据访问接口。实验结果表明，与HDFS和GlusterFS相比，TDFS分别实现了54.38%和57.7%的接口响应速度提升。同时，为了降低湍流大数据的存储开销，设计了基于HDF5的副本延迟压缩机制，相比原有的副本存储方式，节省了34%的存储空间。

关键词
湍流大数据；分布式存储系统；副本延迟压缩；性能优化

中图分类号：TP333 文献标志码：A doi: 10.11959/j.issn.2096-0271.2024046

Research on key technologies for efficient storage and access of turbulent big data

CHENG Wendi[1], ZHANG Xiao[1], PAN Zhaohui[2], ZHAO Youjun[2], SUN Chenguang[1], SHAN Xueqiang[1], JIN Yuzhan[1], ZHAO Xiaonan[1]
1. School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
2. School of Software, Northwestern Polytechnical University, Xi'an 710072, China

Abstract
With the development of measurement techniques and numerical simulation technologies, data-driven turbulence research has become a new approach in this field. In China, several wind tunnel laboratories and supercomputing centers have been established for turbulence simulations, resulting in a substantial collection of turbulence data. However, there is currently no centralized turbulence data management platform in China, which makes it difficult to achieve the exchange and share of the expensive experimental and simulation data. Turbulence data is characterized by its large volume, high

dimensionality, precision and heterogeneity, which present problems in terms of storage, access and management efficiency. A turbulence big data distributed storage system called TDFS was designed, specifically targeting typical flow problems in aviation, aerospace, and marine applications. Considering the access characteristics of turbulence big data, the novel metadata management methods and data access interfaces were designed in TDFS. Experimental results demonstrate that TDFS achieves interface response speed improvements of 54.38% and 57.7% compared with HDFS and GlusterFS, respectively. Additionally, to reduce the storage overhead of turbulence big data, a lazy replication compression mechanism based on HDF5 was designed, resulting in 34% reduction in storage space, compared to the original replication storage approach.

Key words
turbulence big data, distributed storage system, lazy replication compression, performance optimization

0 引言

湍流是自然界中普遍存在的一种三维非定常、有旋的强非线性多尺度流动的流体运动状态。湍流现象存在于众多自然现象和工程应用中，如大气运动、气候变化、海洋环流、飞行器气动力学、能源转换等。随着机器学习和大数据技术的发展，数据驱动的湍流研究已成为理论分析、数值方法和实验技术之外的第四研究范式。得益于大规模科学模拟实验和高精度观测设备的发展，科学数据的生成呈现爆炸式增长的趋势。近年来，研究人员通过精细化的实验测量手段（如粒子图像测速（particle image velocimery，PIV）[1]）和高分辨率数值模拟方法（如直接数值模拟（direct numerical simulation，DNS）、雷诺平均模拟、大涡模拟[2-3]等获得了更细粒度的湍流数据。现代TR-TPIV（time-resolved tomographic PIV）实验通常每分钟生成TB级的数据[4]，湍流数据规模的快速增长早已超过了单机存储的极限。数据驱动的湍流研究可通过对数据进行分析来发现潜在的物理规律，也可验证已有物理模型的正确性。目前国内高精度实验设备和超算中心耗费巨大资金产生的湍流数据，但对其缺乏高效的存储和共享机制。

在美国、日本和西班牙等国家，研究机构已建立了多个湍流数据库[5-11]，并通过网络公开大量的数据，实现了数据的互通和共享。例如美国的约翰霍普金斯大学湍流数据库[9]，目前由9组基础湍流问题的DNS数据集构成，存储量在430 TB以上。该数据库发布后，每年都有100余篇论文利用该数据库的湍流数据进行模型验证。这些数据为研究湍流的基础问题和解决工程的应用问题提供了重要的支持。但是我国的科研人员在访问这些数据库时效率低，且随时可能被限制访问。

近年来，我国加大了对基础研究和超算中心的投入，各大科研机构积累了大量的仿真与实验湍流数据。这些数据的获取成本高，目前仅限于科研机构内部使用，成了多个“数据孤岛”。因此，笔者研发了湍流大数据的存储管理系统，与国内多所高校和科研机构进行合作，取得了面向航空、航天和航海等典型流动问题的中高雷诺数复杂流动问题的数据集，目前已收录了超过100 TB的数据。这些数据将为工程应用导向的数据驱动湍流研究提供有效的支持。**表1**为一些关键部件典型工况的湍流数据集及其特征。

图1为某型号散热器的流体运动情况，湍流数据的模拟在空间上将装备划分为网格，然后在时间上连续计算或测量不同网格的物理量。汽车和飞机的气动模拟需要

表1 典型湍流数据集

数据集	流动问题	雷诺数量级	数据规模	数据特征	访问特征
高超声速飞行器典型外形绕流数据	平板、压缩拐角、圆锥	千万	约87 TB	高维度、高精度、数据量大、多源异构、多变量、多尺度	一次写入多次读取、大量顺序访问、计算复杂、点查询与区域查询结合
民用大飞机翼型绕流数据	二维、准三维翼型	百万至千万	约30~50 TB		
航空发动机典型部件绕流数据	高亚声速压气机、叶栅	十万至百万	约20 TB		

划分为十亿级网格，航空发动机全机模拟需要划分为百亿级网格。为了表示流体随时间的变化，湍流数据通常用短时间间隔的物理量快照进行表示。每个网格，物理量可能包括三维空间的速度、压力、温度和密度等。

湍流数据文件较为庞大（如单个翼型湍流数据文件的大小在50 GB以上）。湍流数据产生后，一般不会再对其进行修改，具有一次写入、多次读取的访问特征。湍流研究利用这些数据进行粒子追踪和场域数据分析，具体到数据文件的查询，表现为指定参数下的点查询或范围查询的形式，涉及对大规模数据集的随机查询、顺序读取和计算。不同研究机构使用的数据格式（如hdf5、dat、plt、mat等格式）差异较大，使用数据时需要阅读数据的格式说明，给数据的共享和高效访问带来了极大的不便。

湍流数据具备独特的数据特征和访问需求，在进行湍流大数据的管理、共享、计算和分析时，如果直接采用HDFS、Ceph、Lustre等通用分布式存储系统会面临以下挑战。

● 在通用分布式存储系统中，大文件被划分为固定大小的数据块，每个数据块都被存储在不同的数据节点上。湍流数据文件通常为二进制格式，在对文件进行分块时，可能恰好在关键位置被截断，导致无法直接读取分块后的文件。基于湍流数据的计算任务通常涉及对文件中整个物理场的运算或多个物理场的联合计算，因此在读取数据时，仍需要合并分块后的数据块来还原完整的文件。

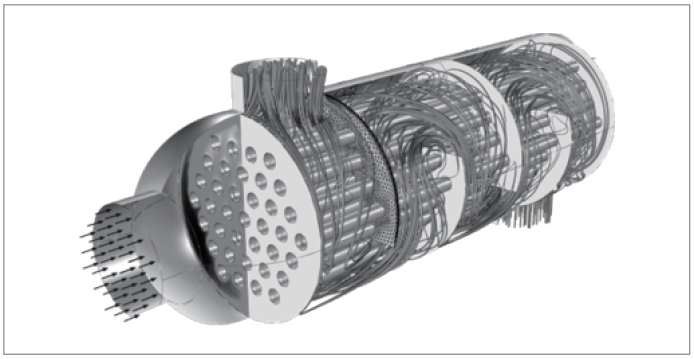

图1 湍流研究对象

● 湍流数据文件通常采用私有二进制格式，以特定的格式和顺序进行存储，其头部保存了数据集的具体描述信息。在访问数据时，需要读取文件头部，以定位数据的具体位置。这种数据格式的组织结构简单，但缺乏清晰的组织结构和标准化的规范，使数据的可读性、可维护性和查询效率有所下降。实验结果表明，当读取指定点和指定区域的数据时，HDF5格式的读取性能比二进制格式读取性能分别提升了59.93%和99.93%。

● 通用分布式文件系统通常采用冗余副本或纠删码机制来保证数据的可靠性和可用性。冗余副本机制带来了额外的存储开销，三副本存储时，其存储空间的利用率仅为1/3，在面对大规模科学数据的爆发式增长时，这会带来显著的存储成本上升问题。而纠删码技术虽然具有高效的冗

余和容错能力，但其编解码过程也会带来额外的计算开销。

为了实现对海量科学数据的高效管理，许多科研机构研发了各类专用的科学数据管理系统。例如，天河二号超级计算机为了解决高性能计算、大数据和人工智能融合背景下的海量科学数据的高效存储问题，提出了分层数据管理、DATS、索引和查询处理等多种优化方案，并设计了新的元数据管理和小文件存储机制[12]。欧洲核子研究组织（CERN）为了存储大型强子对撞机WLCG实验产生的海量数据，建立了开放的海量科学数据存储管理系统EOS（NASA's earth observing system），目前已存储了930 PB的数据供科研人员进行数据共享和协作[13]。Google Earth Engine、Sentinel Hub和ODC（OceanBase）等平台专注于存储和处理地球观测卫星产生的海量地理空间数据，提供了强大的工具和丰富的资源，以支持地球科学研究、遥感影像分析和应用开发[14]。针对海量科学数据的存储需求问题，HDF5、NetCDF和Zarr等自描述文件格式被广泛用于物理、气象和天文等多个领域。其中，HDF5在与NetCDF和Zarr的比较中显示出了更高的输出读取性能[15]，并且具备更丰富的生态系统和更好的跨平台兼容性，更加适用于处理湍流数据的访问场景。此外，为了降低存储成本并提高数据传输效率，研究人员提出了数据压缩、重复数据删除、纠删码、数据异构分层存储等多种方案[16-19]。

综上所述，湍流大数据的存储与管理面临着存储需求大、格式不统一、访问困难、计算任务复杂等挑战。为了支持数据驱动的湍流研究，本文设计并实现了一个面向湍流大数据的高可用、高可靠的分布式数据存储系统。本文的主要贡献如下。

- 设计并实现了一个具有高可用性、高可靠性的面向湍流大数据的专用分布式存储系统TDFS（turbulence data file system），该系统结合湍流大数据的访问特征，对文件存储方式、元数据管理及数据访问接口进行了优化，实现了数据共享、通用算子处理、自定义处理等功能。系统测试结果显示，与开源大数据存储系统HDFS相比，TDFS的接口调用平均响应时间减少了54.38%，相较GlusterFS，TDFS的接口调用平均响应时间减少了57.7%。
- 针对多源数据格式不统一、使用不便的问题，设计了基于HDF5的统一数据存储规范，将多样化、异构的湍流数据集统一转化为HDF5格式进行存储和管理，提高了湍流大数据的存储和处理效率。
- 针对多副本空间利用率低的问题，结合湍流数据的只读特性，提出了副本延迟压缩方案，解决了冗余副本机制带来的存储效率较低的问题。测试结果表明，与传统的副本存储策略相比，该方案节省了34%以上的存储空间。

1 湍流大数据存储优化

针对湍流大数据存储存在的数据集成困难、数据访问低效和存储效率低等问题，本文以HDFS为原型设计了一个湍流大数据分布式存储系统TDFS。为了实现对大规模湍流数据的高效管理、访问与共享，TDFS基于湍流大数据的结构和特征，设计了新的元数据组织方式和数据访问接口。同时，为了降低湍流大数据的存储开销，本文提出了一种基于HDF5的副本延迟压缩优化机制。

1.1 湍流大数据存储系统架构

如**图2**所示，TDFS采用了主从架构，

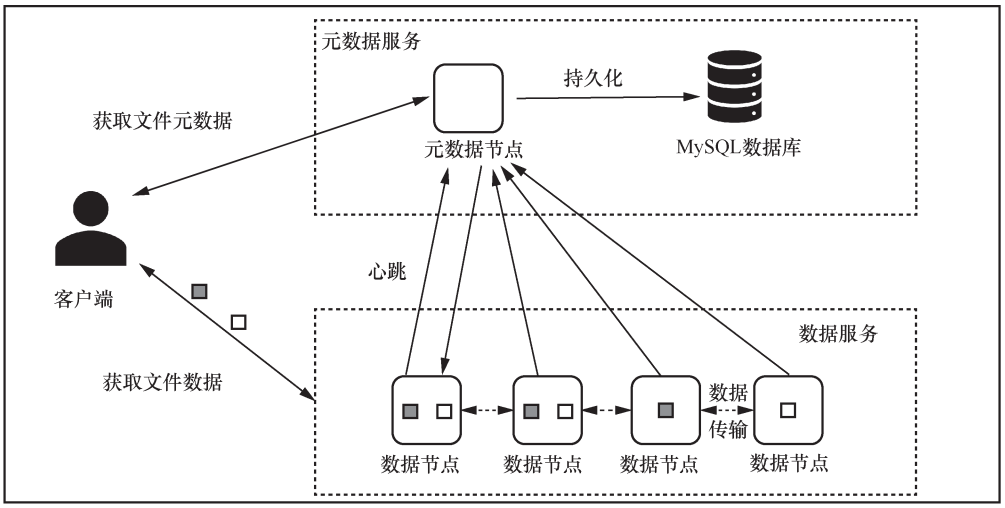

图 2 TDFS 架构

由客户端、元数据节点和数据节点3类节点组成。为了保证存储系统的可靠性和可用性，TDFS采用了三副本冗余的存储策略，当某个数据节点发生故障时，可以通过其他副本快速恢复数据。考虑到湍流数据具有频繁读取的访问特性，副本机制可以满足并行读取和故障快速恢复的需求，加快系统的响应速度和提升读取性能。纠删码机制在数据恢复时需要复杂的编解码过程，会引入额外的计算开销和延迟，尤其是在频繁读取场景下，这些额外的计算开销可能会成为瓶颈，影响系统的响应速度和读取性能。TDFS采用RESTful架构和HTTP通过标准化的接口进行节点间通信，实现了松耦合和可扩展性。

主流分布式存储系统大多采用分块机制来进行大文件存储，对于湍流数据的计算及访问，分布式存储系统内部需要频繁的网络和磁盘I/O来重建流场数据，严重影响系统性能。为了实现对湍流数据集的高效访问，解决节点间网络传输带来的开销问题，TDFS并未对文件进行分块存储，而是将数据集存储为单独的文件，并维护了每个数据集的元数据信息，将文件均匀地存储在集群的多个数据节点中。相比传统的分布式文件系统，TDFS在数据访问和管理方面具有更灵活和可扩展的数据处理能力。

1.2 元数据组织优化

湍流数据在多个时间步进行采样，生成的数据具备一次写入、多次读取的访问特征，因此其元数据信息具备不可变性。此外，湍流数据文件以大文件形式存储（通常为GB级），系统整体的元数据规模较小。TDFS在元数据节点中维护了层次化的湍流数据集结构和描述信息，采用了MySQL与Redis来缓存元数据，将元数据信息持久化存储到MySQL数据库中。为了加快访问速度，TDFS在缓存过程中也维护了系统运行需要的元数据信息，并提供高效的访问接口。按照单个文件的元数据平均占用300个字节来计算，存储1亿个文件大约需要30 GB内存。现阶段单机Redis能够满足湍流数据集元数据存储的需求，随着数据规模的扩大和并发请求的增加，可以使用Redis集群进行水平扩展来提升性能、扩大容量。

在湍流模拟中，每个时间步都会产生一个数据文件，在处理湍流文件时，需要通过时间步信息来索引数据，因此对于每种类型数据集下的所有湍流数据文件，TDFS在缓存时额外维护了不同时间步到数据文件名的实时映射关系。此外，时间步通常是一个递增的浮点数且具备不可变性，因此TDFS采用了键值对的形式来组织和存储文件信息。

TDFS元数据节点在缓存过程中维护了文件系统的目录树，每个文件和目录项统一用Inode类的实例表示，目录项和文件以哈希表的形式组织，哈希表以文件Inode号为key、实例对象为value，形成了树状结构。内存中的Inode实例对象对应数据库中inodes表中的记录，Inode对象之间的引用关系用dentries表来记录。如**图3**所示，flatplate目录下存储了一个名为23800.h5的文件，那么flatplate和23800.h5在缓存中有两条Inode记录，两者的ID分别为2和6，dentries表中则会有一条<2,6>格式的记录，表明ID为6的文件存储在ID为2的目录下。

1.3 面向湍流数据的数据集成和访问处理优化

湍流数据的生成和采集是湍流研究的关键步骤，目前国内知名的湍流研究机构具备生成大规模湍流数据的能力，然而不同机构使用的数据格式和标准各异，大部分湍流数据采用二进制文件格式进行存储，每次读取指定点时都需要手动计算偏移量，这给数据的共享和访问带来了极大的不便。为了有效地整合与统一存储多样化的湍流数据集，TDFS将写入的数据统一转化为标准的HDF5格式进行存储，并保留文件的元数据描述信息，从而确保数

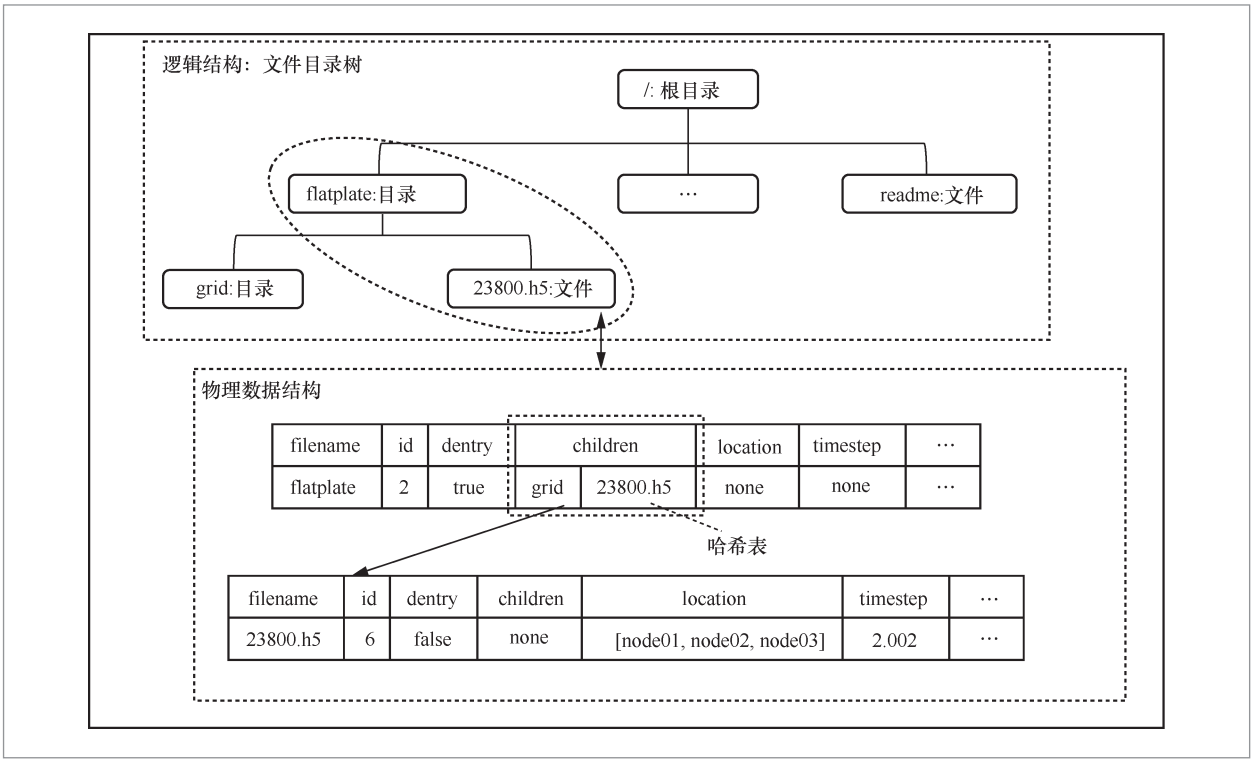

图 3 元数据节点文件中元数据的逻辑结构和物理结构

据的准确性和可用性。如**图4**所示，二进制格式的湍流源文件可以转换为层次化、标准化的HDF5格式，这种格式不仅包含丰富的元数据信息，还支持高效存储和快速访问，能够有效地提高数据处理和分析的效率。本文以Ma2.25超声速平板湍流边界层的瞬时场数据为例，对源文件进行了转换。测试结果表明，当读取指定点数据和范围数据时，HDF5格式的读取性能比二进制格式的读取性能分别提升了59.53%和99.93%。

TDFS中存储了上百TB的湍流数据，由于不同用户对数据的访问需求不同，系统提供了文件共享、通用算子处理和自定义运算3种基本的访问方式，如**图5**所示。

部分用户不具备存储大容量数据的条件，或者希望尽快地验证和处理数据。TDFS提供了通用算子和自定义运算两个功能。通用算子功能将流场数据查询、数据计算、统计分析等常用的处理过程抽象为通用的算子供用户调用。用户可以直接调用特定算子对文件进行高效处理，只需要向系统发送获取数据的详细请求，即可精准地提取所需数据，并获得计算结果，借助这个功能，用户无须下载庞大的数据集就可以实现实时的数据分析。根据数据量的大小，TDFS将通用算子功能划分为单点、区域和全量流场计算3种类型。单点计算是针对单个点进行的查询与计算操作。区域计算涉及对一个具体区域内的数据的计算，需要读取该区域周围立方体的数据，并通过差分法等算法进行计算，例如计算梯度、海森矩阵以及拉普拉斯矩阵等。全量流场计算涉及对整个流场的数据的计算。此外，区域计算与全量流场计算的读取和计算类型取决于具体的计算任务。由于大部分计算任务需要等待所有数据读取完成后才能开始计算，采用了串行化的调度方案。自定义运算功能则支持对多个文件的并行访问，用户可以通过上传自定义代码来发送计算任务请求，从而实现快速、在线的数据处理和计算，最终获得处理后的结果。

1.4 副本延迟压缩优化

为了解决副本机制带来的存储效率下降的问题，本文基于惰性复制的思想，提出

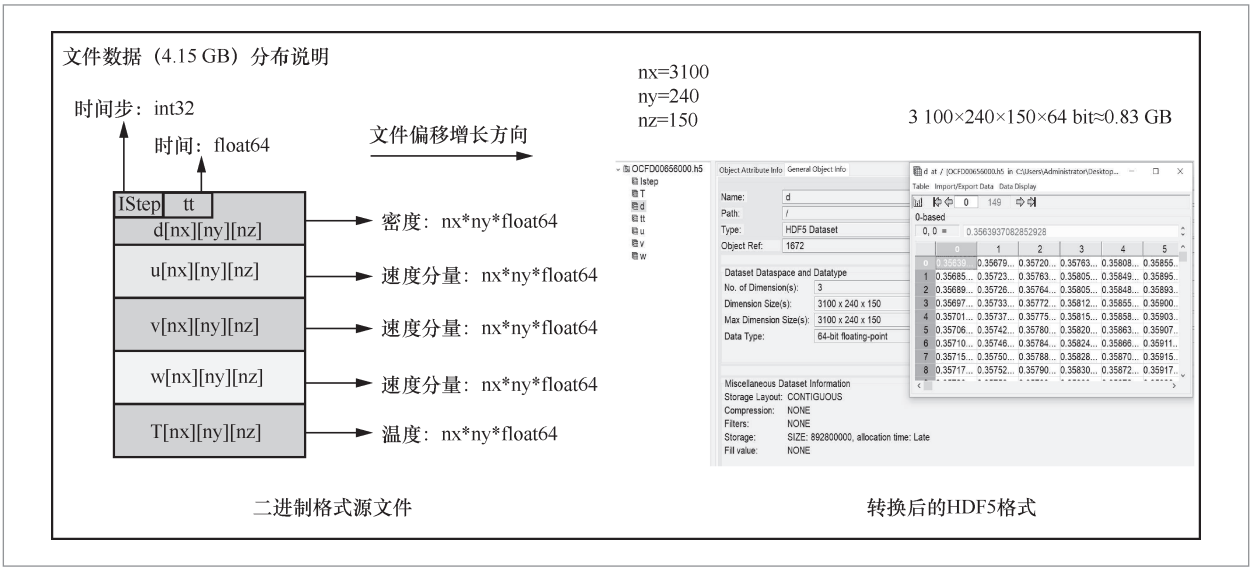

图 4 湍流数据存储格式对比

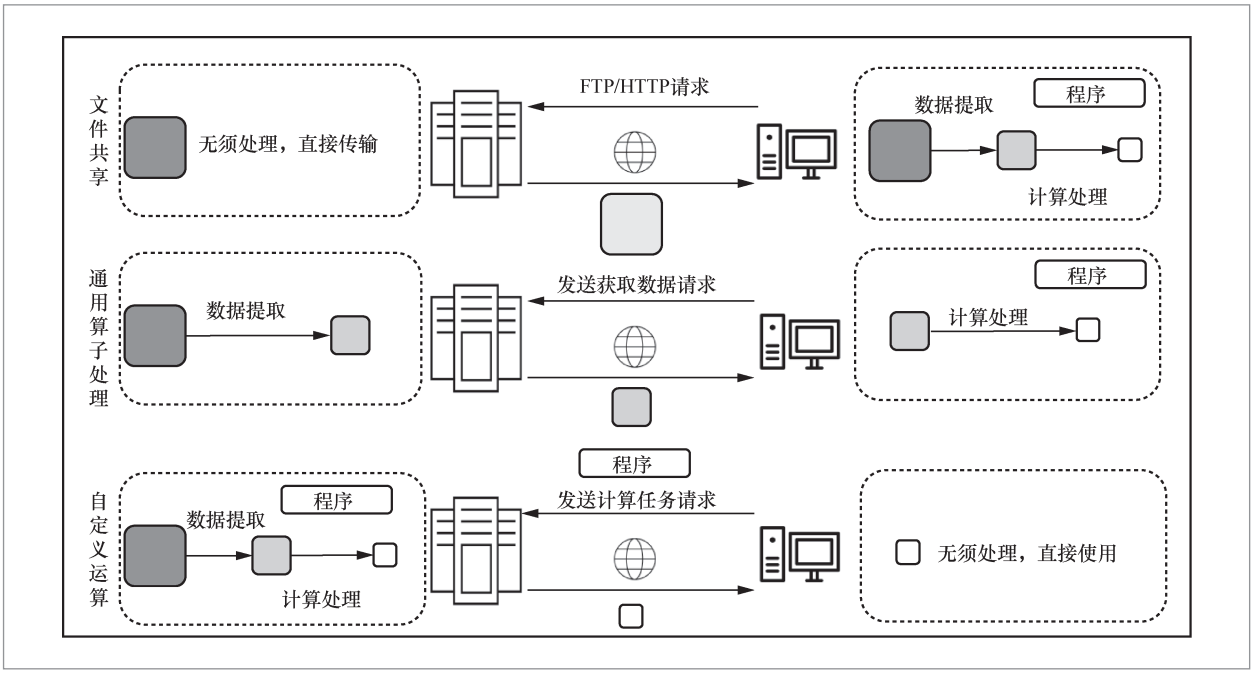

图 5 TDFS 提供的 3 种访问方式

了一种基于延迟压缩（lazy compression）的存储效率优化方案。该方案的核心思想是在集群中仅保留一个完整文件，将其余的副本文件进行压缩存储。同时，为了避免副本压缩对写入性能的影响，将压缩延迟至所有副本写入完成后，再异步执行。

TDFS延迟压缩方案的写入数据流程如图6所示。首先，客户端向元数据节点发送写请求获得文件所在的数据节点，并建立数据流传输管道；然后，客户端开始向数据节点写入数据。数据节点成功写入副本后，将该数据的副本添加到异步任务队列中。同时，后台的异步任务模块会动态检测队列中是否存在待压缩的副本。一旦发现待压缩的副本，空闲线程将异步压缩该数据副本。由于HDF5支持动态重新压缩，重新压缩可以直接在现有文件上执行，无须删除原始文件。当异步压缩完成后，调用数据节点的相关接口来更新文件和其他元数据。在异步压缩后，同一文件的两个副本被压缩存储，而一个副本按照原始格式被存储。

传统的分布式系统（如HDFS），会依据以下优先级来选择就近的可用数据节点：同一节点或同一机架的不同节点、同一数据中心不同机架的节点、不同数据中心的节点。为了实现最佳的文件读取性能，本文结合延迟压缩机制设计了新的副本选取策略。延迟压缩机制将存储的部分副本进行压缩存储，客户端在下载文件时有两种选择：下载原始副本文件（方案1）或下载压缩副本文件再解压缩（方案2）。在延迟压缩机制中，需要综合考虑网络传输和文件解压缩带来的影响，寻求合适的平衡点。

假设TDFS下载压缩副本的时间为t_{cf}，下载完整副本的时间为t_{ucf}，客户端解压缩文件的时间为t_{dc}。为了最大化系统读取性能，系统增加了以下规则：若系统下载压缩副本节省的时间$t_{ucf}-t_{cf}$小于解压缩时间t_{dc}，那么说明系统的瓶颈在于解压缩，这时

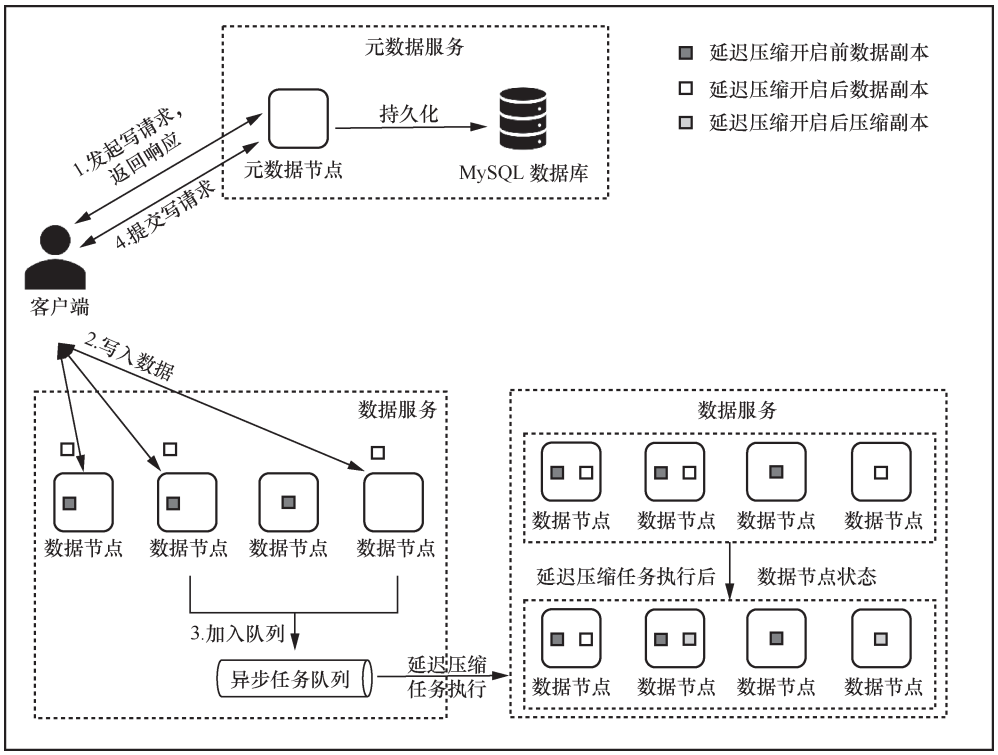

图 6　TDFS 延迟压缩方案的写入数据流程

应选择方案1。若系统下载压缩副本所节省的时间$t_{ucf}-t_{cf}$大于解压缩时间t_{dc}，那么说明系统的瓶颈在于网络传输，这时应选择方案2。

数据压缩的效果取决于原始数据集、数据格式和采用的压缩算法，不同压缩算法的压缩比、压缩速度和解压缩速度等指标有较大的差异。湍流数据文件以时间步长为间隔进行采样，但由于每个时间步长对应不同的文件，单个文件中数据的连续性和相关性不足，因此并未使用差分编码的方案进行压缩。HDF5是一种用于存储和管理大规模科学数据集的文件格式，也是湍流文件的常用存储形式。对于HDF5格式文件的压缩，目前有两种可行的方案：一种是直接湍流文件进行压缩；另一种是对HDF5文件执行数据集级别的压缩。

为了获取在湍流数据集上使用不同压缩算法的效果，本文在原始大小为8.1 GB的各向同性湍流HDF5数据集上，对无损压缩算法Deflate、Bzip2、LZMA、LZ4、Snappy及Zstd进行了测试。**图7**所示为各压缩算法的压缩时间、解压缩时间、压缩后的文件大小。通过子图(a)和(b)可知，Zstd和LZ4算法的压缩速度和解压缩速度较快，特别是Zstd的解压缩性能很好。由子图(c)可知，压缩后的文件大小相差不大，在3.3 GB附近波动。

为了验证湍流数据集使用不同HDF5数据集级别压缩算法的效果，本文对常用的Gzip、LZF、Bitshuffle、Blosc、Blosc2等12种过滤器插件进行了测试，如**图8**所示。实验结果表明，Lzf、Bitshuffle、Blosc、Blosc2、LZ4和Zstd过滤器在插件综合性能方面表现更好，且性能差异不

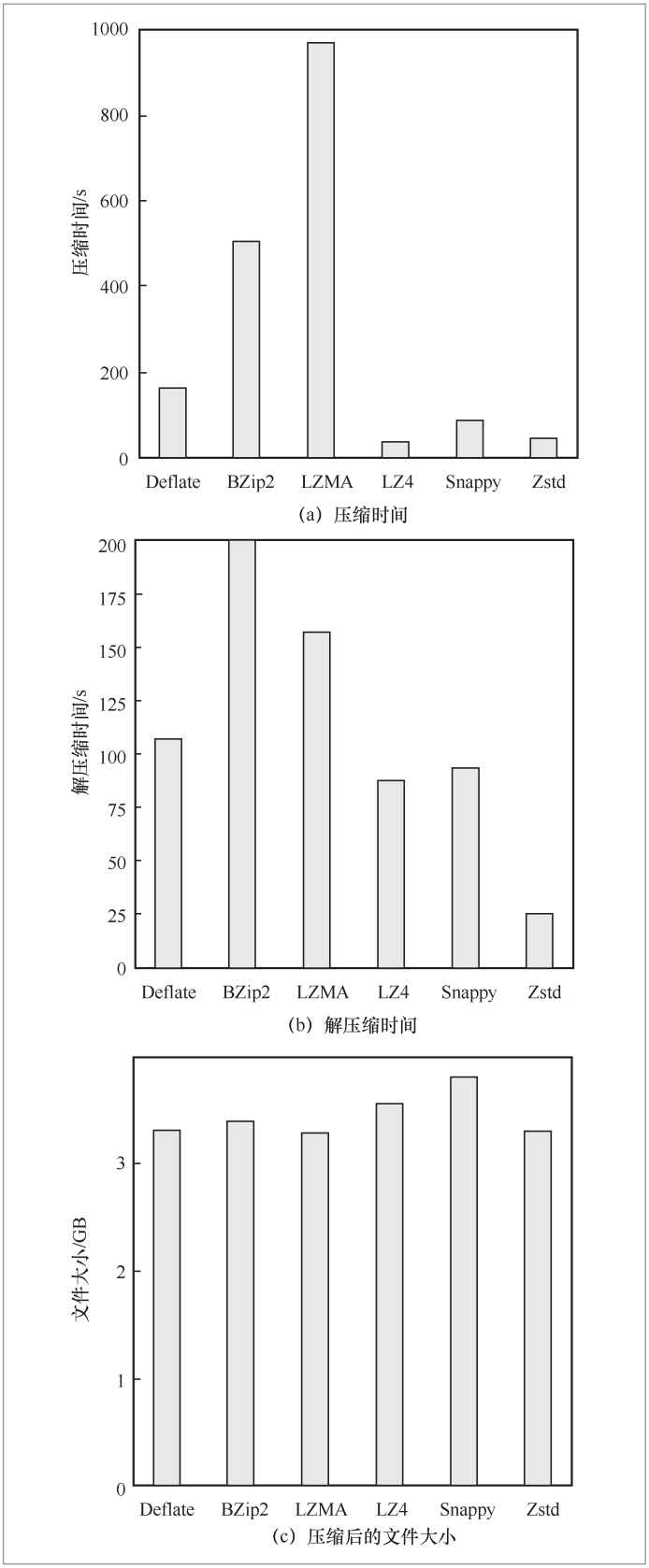

图 7 各压缩算法在各向同性湍流数据集下的性能

大。在压缩速度和解压缩速度方面，Zstd过滤器插件略优于其他过滤器插件。而在压缩比方面，SZ3过滤器插件的表现最好。

为了对比上述两种方案的效果，本文选取3种不同类型的湍流数据集进行测试（3种典型条件下的槽道流与各向同性湍流文件，原始大小分别为0.125 GB、1.5 GB和8.1 GB）。如**图9**所示，子图（a）、（b）、（c）分别代表了压缩时间、还原时间和压缩后的文件大小。其中，还原时间是指从读取文件数据进行解压缩到全部原始数据被加载到内存的时间，对“基于文件压缩”的还原时间指的是压缩后的文件被解压缩为原始文件，再从原始文件读取流场数据到内存中的时间，“基于HDF5接口压缩”的还原时间指的是从完成解压缩数据集后到加载至内存中的时间长度。

由**图9**可知，直接对文件进行压缩和基于HDF5接口进行压缩，两者的压缩速度和压缩率效果相差不大，但是在解压缩速度方面，后者的性能远优于前者。这是因为HDF5格式可以在数据集级别进行压缩，读取数据时可以仅解压缩当前的数据集，而不必解压缩整个文件后，再去读数据。同时，相比将整个文件进行压缩，基于HDF5的数据集级别的压缩方案可以根据数据的特性和需求进行灵活的设置，具备更大的灵活度。

本文提出的延迟压缩机制将压缩操作与写入操作进行解耦，将压缩过程放入后台异步执行，从而避免了实时压缩对写性能的影响。为了减小对读取性能的影响，需要确保解压缩的速度足够快。综合上述实验结果可以发现，相较其他压缩算法，基于HDF5的Zstd压缩方案不仅可以维持较低的压缩率，还表现出了较快的解压缩速度。因此，将其作为后续的测试方案，进一步验证本文提出的副本延迟压缩方案在实

际应用中的效果。

2 评估与测试

2.1 测试环境

本文搭建了一个面向湍流大数据的分布式存储系统——TDFS，包括1个元数据节点、4个数据节点，每个数据节点配备了10个8 TB的机械硬盘。集群现有的湍流数据存储量约为100 TB，且在不断扩充。本文所有测试的软件、硬件环境分别见**表2**、**表3**。

2.2 TDFS存储系统性能测试

湍流数据管理平台支持文件共享、通用算子和自定义运算3个主要功能。为了评估TDFS的表现，本文将TDFS与通用的分布式存储系统HDFS、GlusterFS进行对比，选用各向同性湍流数据集分别测试各个系统的接口响应时间，如**图10**所示。

对于文件共享功能，本文通过测试3个分布式文件系统的调用Get接口的响应时间来评估它们的性能。结果显示，3个系统的响应时间差异不大。这是因为文件共享的性能主要受限于网络速度，而不是底层文件系统。

对于通用算子功能，由于HDFS和GlusterFS无法根据时间步来索引文件地址，同时为了避免网络传输对接口性能的影响，笔者在本地节点上对各向同性湍流数据集进行400次随机点查询测试，计算接口主要测试了点查询（如GetVelocity）和区域计算（如GetVelocity gradient）两种接口。结果显示，相比HDFS，TDFS通用算子接口的平均响应时间减少了54.38%，相比GlusterFS减少了57.7%，TDFS的表现明显优于HDFS和

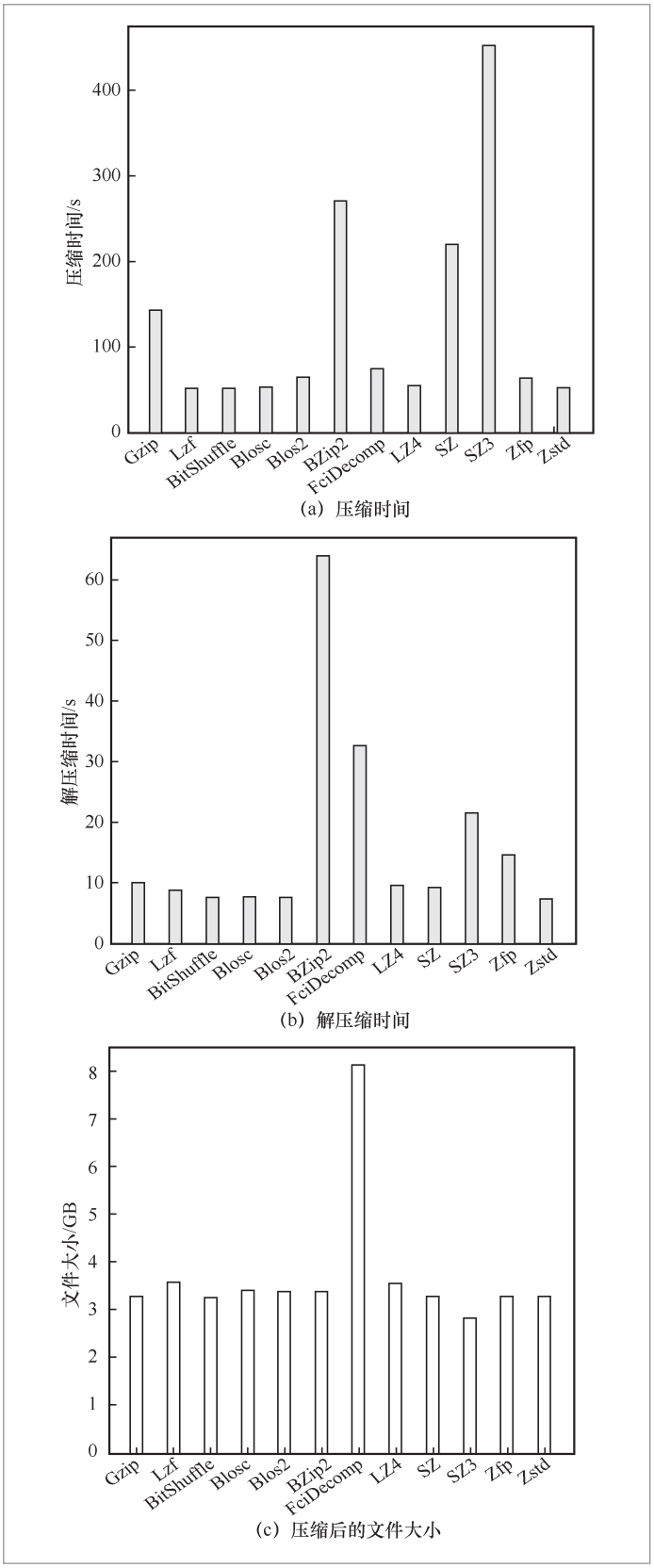

图 8 HDF5 过滤器插件在各向同性数据集下的表现

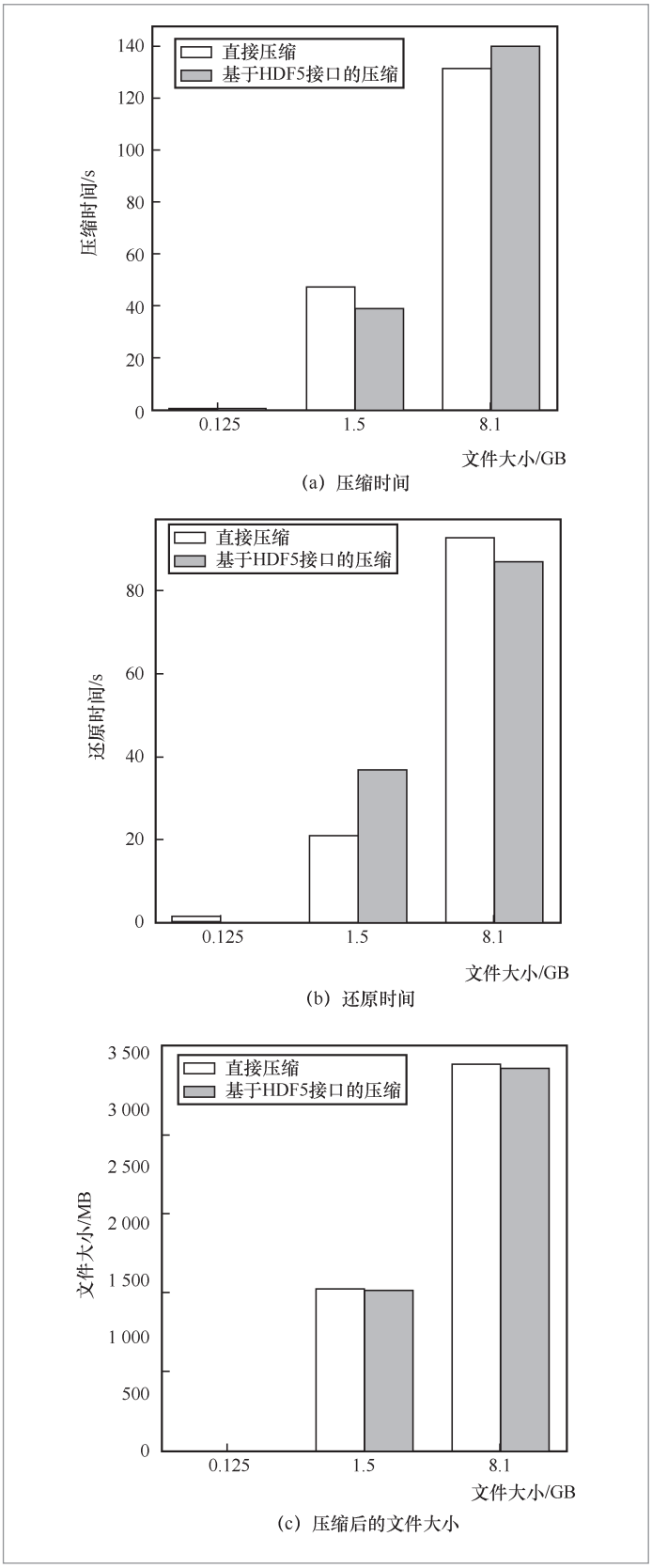

图 9 直接压缩与基于 HDF5 接口压缩的性能

GlusterFS。

对于自定义运算功能，对各向同性湍流数据集下的速度场进行计算，并将结果输出。结果显示，TDFS的平均响应时间比HDFS的减少了39.9%，比GlusterFS的减少了44.56%。

对于通用算子和自定义运算系统的性能测试，TDFS明显优于HDFS和GlusterFS，主要是因为TDFS可以充分利用数据本地化的优势，后两者在多数情况下必须通过网络传输数据。为了分析TDFS的资源消耗情况，本文对比了3种不同分布式文件系统的节点资源变化情况，如**图11**所示。可以看出，HDFS和GlusterFS需要通过频繁的网络通信来获取数据，三者在计算时对内存、CPU资源的使用情况没有明显的差别。

2.3 基于TDFS的延迟压缩方案的测试

为了评估基于TDFS的延迟压缩方案的性能，本文在TDFS上执行了基于HDF5的Zstd压缩方案，并对比了在应用该方案前后系统的响应时间与压缩后的数据大小的变化。本文根据不同类型的数据集，按照文件数量等比例进行划分，并从中随机选择了测试数据，**图12**所示为测试对比结果。

图12中原始写入方案指的是TDFS的原始三副本写入方案，延迟压缩方案指的是优化后的延迟压缩方案。由子图(a)可以看出，延迟压缩方案对系统吞吐量和延迟的影响较小。由子图(b)可知，在系统稳定后，延迟压缩方案平均降低了34%的存储空间。

延迟压缩方案会对文件共享、通用算子计算和自定义运算3种功能产生影响。对于文件共享功能，将基于HDF5的文件的内部数据集接口进行压缩后，依然能够使用

HDF5接口进行操作，并且相比下载原始文件，压缩后的数据的下载速度更快。对于自定义运算功能，在使用到压缩后文件时需要经过解压缩步骤，但由于自定义运算是非实时的任务，解压缩带来的额外开销不会对系统的整体性能产生显著的影响。

为了评估延迟压缩方案对通用算子功能的影响，本文在各向同性湍流和槽道流文件上对不同计算接口进行测试，**图13**所示为原始HDF5文件和压缩后的HDF5文件对应的3种计算的响应时间。子图（a）为各向同性湍流中速度场文件（文件大小为8.1 GB）的测试结果，子图（b）为槽道流中速度场文件（文件大小为1.5 GB）的测试结果。可以看出，对于单点和全量流场的计算，TDFS对压缩后HDF5文件的处理性能会降低，而对于区域类型的计算，基于HDF5压缩文件进行计算反而会提高性能。

为了进一步验证数据量对区域计算性能的影响，笔者还在不同数据集下测试了读取不同数量的数据点到内存中所需要的时间，读取数据点的数量与耗用时间的关系如**图14**所示。可以看出，在单点读取时，两者的差距较小，但是在对多点区域进行读取时，压缩后的HDF5文件所需的时间明显少于原始HDF5文件，获得了较大的性能提升。

综上所述，本节在TDFS中实现了基于HDF5的延迟压缩方案的测试，并对优化后的存储优化效果进行了测试，结果显示，该方案平均节省了34%的存储空间。同时，评估了该机制对于文件共享、通用算子计算和自定义运算3种主要功能的影响，发现读取压缩副本时，数据量的大小会对文件的读取性能产生影响，尤其是在进行通用算子计算时，将读取区域数据的范围查询请求调度至压缩副本所在的节点，可以获得一定的性能提升。

表 2　软件环境

名称	版本
操作系统	Ubuntu 20.04.6 LTS
Linux内核	5.4.0
Python	3.7.13
Django	3.2
Nginx	1.18.0
MySQL	8.0.35

表 3　硬件环境

名称	型号
主板	ASUS PRIME B660M-K D4
CPU	12th Gen Intel(R) Core(TM) i7-12700
内存	Kingston 99P5779-001.A00G, 16 GB
硬盘	KINGSTON SNV2S, 500 GB HGST HUS728T8TALE6L4, 8 TB × 10
网卡	RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller

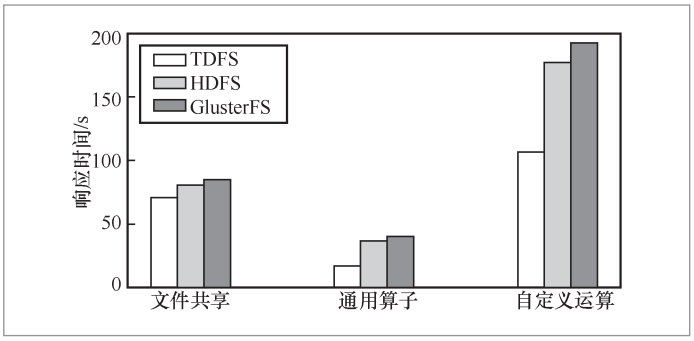

图 10　调用接口在不同存储系统下的响应时间

3 结束语

本文旨在解决湍流大数据研究中存在的数据存储管理问题，为国内外湍流研究机构提供高效、可靠、便捷的数据共享及管理平台。与传统的分布式存储系统相比，本文设计的湍流数据存储系统TDFS

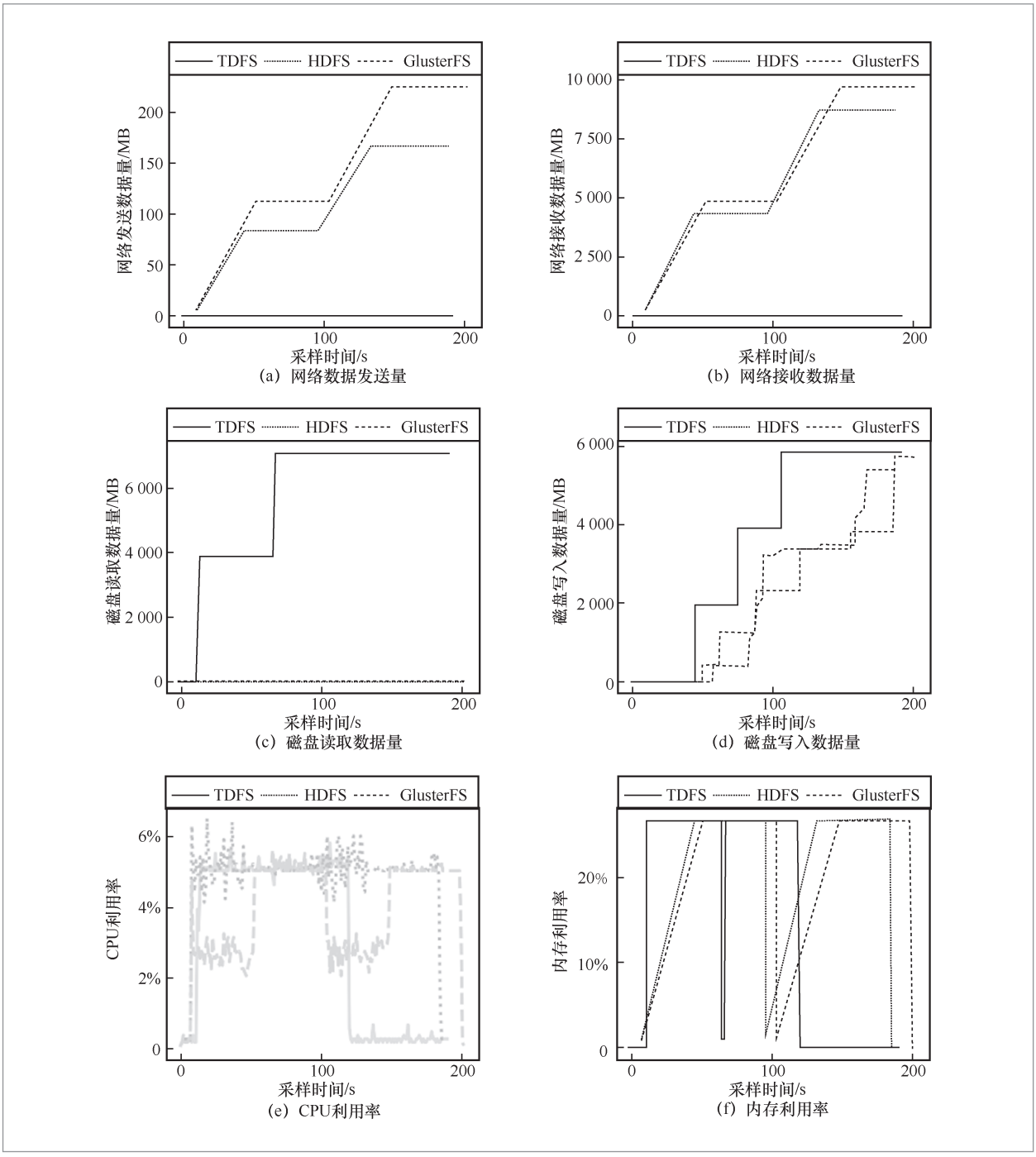

图 11 不同存储系统的资源消耗

采用了基于HDF5的湍流数据集存储方案，并设计了新的元数据组织方式，同时针对冗余副本方案带来的空间利用率降低的问题，提出了副本延迟压缩方案。TDFS为湍流数据访问提供了高效的数据共享、通用算子和自定义运算功能。实验结果表明，在通用算子计算场景下，TDFS的平均响应时间相较HDFS和GlusterFS分别减少

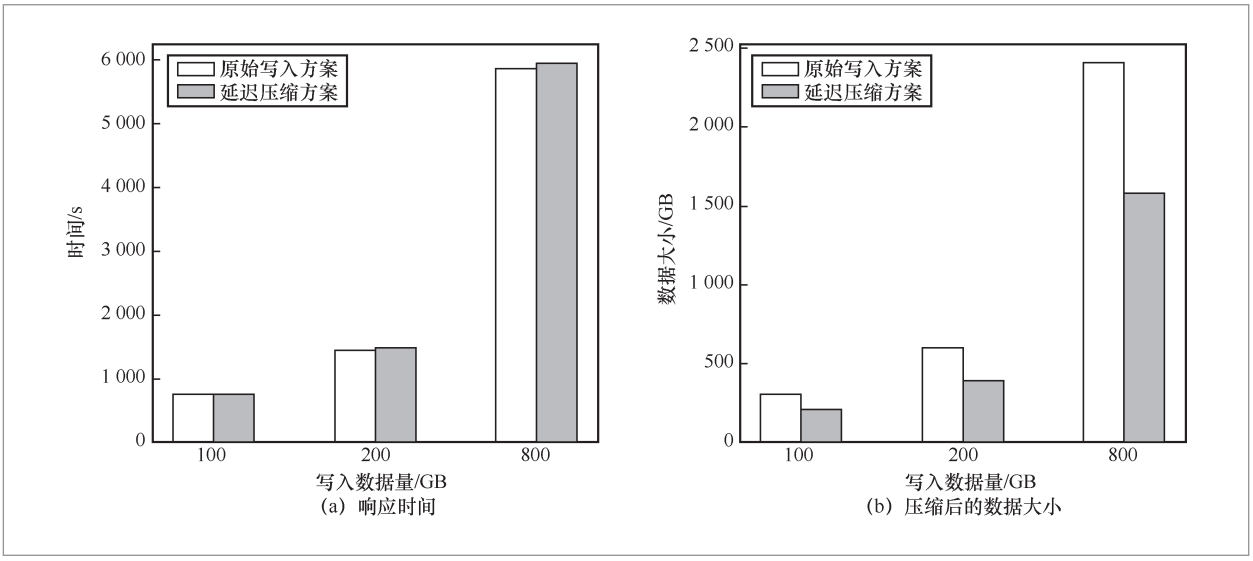

图 12 系统响应时间与压缩后的数据大小

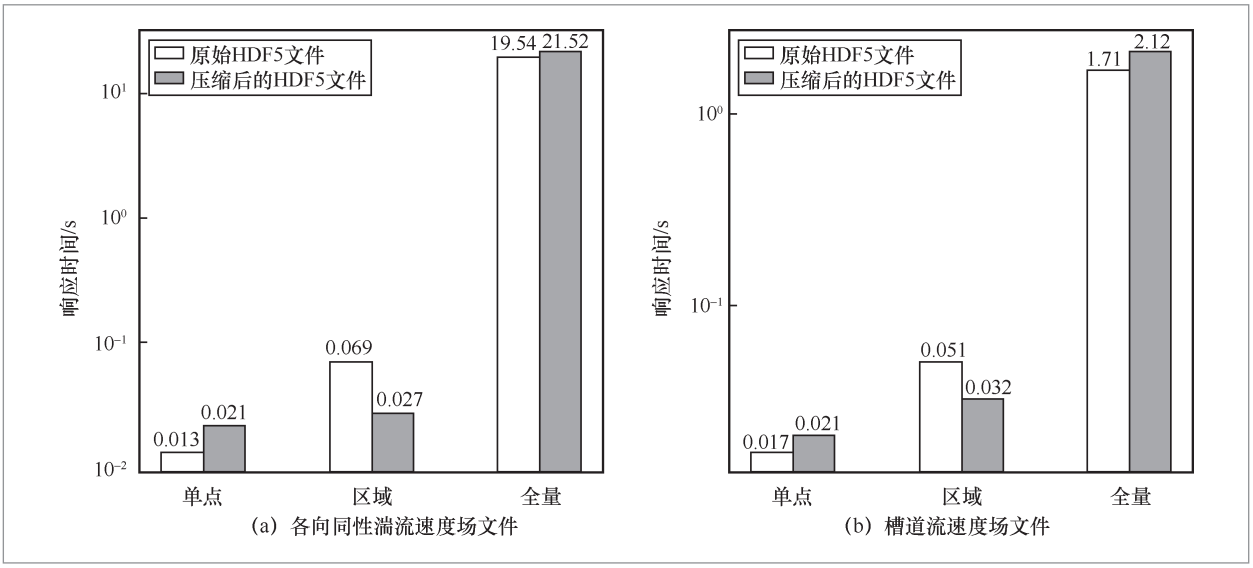

图 13 各向同性湍流文件和槽道流文件对不同接口的响应时间

了54.38%和57.7%。本文提出的延迟压缩方案与传统的副本存储方式相比，平均节省了34%的存储空间。

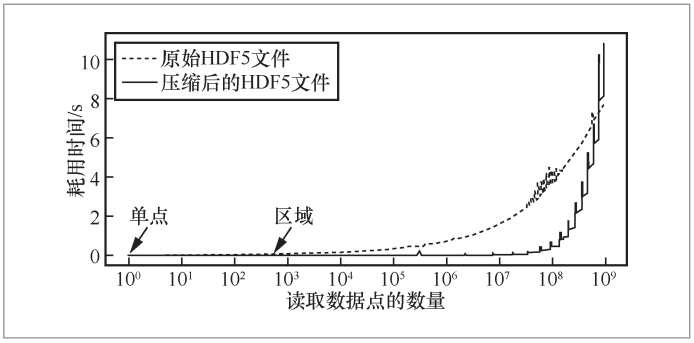

图 14 读取数据点的数量与耗用时间的关系

参考文献:

[1] ELSINGA G E, SCARANO F, WIENEKE B, et al. Tomographic particle image velocimetry[J]. Experiments in Fluids,

2006, 41(6): 933–947.

[2] 李新亮. 高超声速湍流直接数值模拟技术[J]. 航空学报, 2015, 36(1): 147–158.
LI X L. Direct numerical simulation techniques for hypersonic turbulent flows[J]. Acta Aeronautica et Astronautica Sinica, 2015, 36(1): 147–158.

[3] 王圣业, 邓小刚, 董义道, 等. 面向工程湍流的高精度数值方法[J]. 航空学报, 2023, 44(15): 528728.
WANG S Y, DENG X G, DONG Y D, et al. High-order numerical methods for engineering turbulence simulation[J]. Acta Aeronautica et Astronautica Sinica, 2023, 44(15): 528728.

[4] MENEVEAU C, MARUSIC I. Turbulence in the era of big data: recent experiences with sharing large datasets[M]//Whither Turbulence and Big Data in the 21st Century? . Cham: Springer, 2017: 497–507.

[5] WRAY A A. A selection of test cases for the validation of large-eddy simulations of turbulent flows[J]. AGARD Advisory Report, 1998, 345.

[6] SILLERO J A, JIMÉNEZ J. Public dissemination of raw turbulence data[M]//Whither Turbulence and Big Data in the 21st Century? . Cham: Springer, 2017: 509–515.

[7] RUMSEY C L. Turbulence modeling verification and validation[C]//Proceedings of 52nd Aerospace Sciences Meeting. Maryland: ARC, 2014: 0201.

[8] LI Y, PERLMAN E, WAN M P, et al. A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence[J]. Journal of Turbulence, 2008, 9: 1–29.

[9] KANOV K, BURNS R, LALESCU C, et al. The johns Hopkins turbulence databases: an open simulation laboratory for turbulence research[J]. Computing in Science & Engineering, 2015, 17(5): 10–17.

[10] TOWNE A, DAWSON S T M, BRÈS G A, et al. A database for reduced-complexity modeling of fluid flows[J]. AIAA Journal, 2023, 61(7): 2867–2892.

[11] CHEN Z, ZHANG J B, LEE C H. Direct numerical simulation of the turbulent MHD channel flow at low magnetic Reynolds number for electric correlation characteristics[J]. Science China Physics, Mechanics and Astronomy, 2010, 53: 1901–1913.

[12] LU Y T, CHENG P, CHEN Z G. Design and implementation of the Tianhe-2 data storage and management system[J]. Journal of Computer Science and Technology, 2020, 35(1): 27–46.

[13] PETERS A J, SINDRILARU E A, ADDE G. EOS as the present and future solution for data storage at CERN[J]. Journal of Physics: Conference Series, 2015, 664(4): 042042.

[14] GOMES V C F, QUEIROZ G R, FERREIRA K R. An overview of platforms for big earth observation data management and analysis[J]. Remote Sensing, 2020, 12(8): 1253.

[15] AMBATIPUDI S, BYNA S. A comparison of HDF5, zarr, and netCDF4 in performing common I/O operations[EB]. arXiv preprint, 2022, arXiv: 2207.09503.

[16] ZHANG X, WANG L, HUANG Z J, et al. ConeSSD: a novel policy to optimize the performance of HDFS heterogeneous storage[C]//Proceedings of the 2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). Piscataway: IEEE Press, 2022: 876–881.

[17] JAYASANKAR U, THIRUMAL V, PONNURANGAM D. A survey on data compression techniques: from the perspective of data quality, coding schemes, data type and applications[J]. Journal of King Saud University – Computer and Information Sciences, 2021, 33(2): 119–140.

[18] MOHAMED S M A, WANG Y L. A survey

on novel classification of deduplication storage systems[J]. Distributed and Parallel Databases, 2021, 39(1): 201-230.
[19] CHINIAH A, MUNGUR A. On the adoption of erasure code for cloud storage by major distributed storage systems[J]. EAI Endorsed Transactions on Cloud Systems, 2022, 7(21): 170955.

作者简介

程文迪(1995-),女,西北工业大学计算机学院博士生,主要研究方向为分布式存储系统设计与优化。

张晓(1978-),男,西北工业大学计算机学院教授,主要研究方向为分布式存储系统设计、评测、仿真与优化。

潘兆辉(2000-),男,西北工业大学软件学院硕士生,主要研究方向为分布式存储系统优化。

赵友军(1998-),男,西北工业大学软件学院硕士生,主要研究方向为大数据存储、分布式文件系统设计。

孙晨光(1999-),男,西北工业大学计算机学院硕士生,主要研究方向为分布式存储系统设计。

单学强（1999– ），男，西北工业大学计算机学院硕士生，主要研究方向为分布式存储系统设计。

金雨展（2000– ），女，西北工业大学计算机学院硕士生，主要研究方向云计算资源调度、云资源预测。

赵晓南（1979– ），女，博士，西北工业大学计算机学院副教授，主要研究方向为分布式文件系统管理与优化、存储系统资源管理与智能配置、云存储系统性能建模与性能优化。

收稿日期： 2024-04-18
通信作者： 赵晓南，zhaoxn@nwpu.edu.cn
基金项目： 国家自然科学基金项目（No.92152301）
Foundation Item: The National Natural Science Foundation of China (No.92152301)