XHDF5：面向 HEPS 的高性能 HDF5 数据远程访问系统

冯时超[1]，程耀东[2,3*]，程垚松[2]

1．郑州大学，计算机与人工智能学院，河南省 郑州市 450001
2．中国科学院高能物理研究所 ，计算中心，北京市 100049
3．中国科学院大学，北京市 100049

摘　要：【目的】为满足异地用户对高能同步辐射光源（HEPS）海量数据的访问需求，提出了一种高效且具备良好适应性的基于 HDF5 数据格式的远程访问解决方案。【应用背景】HEPS 每年将产生海量科学数据，以 HDF5 作为统一的数据格式，采用“数据集中存储，异地分布处理”的计算模式，用户的计算任务无需提前下载数据即可无缝地运行在异地的算力中心上。【方法】通过对 HEPS 数据中心的存储现状和目前数据访问手段的分析，采用 HDF5 VOL 与 XRootD 实现对 HDF5 数据的透明访问，设计并实现了 XHDF5 系统。该系统采用了并发访问、数据压缩等方式提高访问性能。【结果】通过多项技术测试、性能对比实验以及大规模数据处理测试，XHDF5 系统在远程访问 HDF5 文件时展现出卓越的性能和高度的稳定性。与传统的基于文件系统挂载的访问方法和 H5serv 技术相比，XHDF5 系统在高网络延迟的环境下表现尤为突出。【结论】XHDF5 系统能够高效地支持跨区域的数据共享与协同处理，为科研人员提供一个稳定可靠的高性能数据访问环境，助力科学研究的顺利开展。

关键词： XRootD；HDF5；HEPS；NFS；远程数据访问

XHDF5: a high-performance HDF5 remote data access system for HEPS

FENG Shichao[1], CHENG Yaodong[2,3*], CHENG Yaosong[2]

1．School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, 450001, China
2．Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, 100049, China
3．University of Chinese Academy of Sciences, Beijing, 100049, China

Abstract:【Objective】In order to meet the demand of remote access to massive data generated by high energy synchrotron radiation source (HEPS), an efficient and adaptable solution based on HDF5 data format was proposed. 【Background】The HEPS is expected to produce a tremendous volume of scientific data annually. By adopting HDF5 as the unified data format and employing the "centralized data storage, remotely distributed processing" computing paradigm, users are no longer required to download data in advance. Instead, computing tasks can be seamlessly run on the remote computing center. 【Methods】By analyzing the current storage infrastructure of the HEPS data center and existing data access methods, the XHDF5 system has been designed and implemented. Leveraging HDF5 VOL and XRootD technologies, this system facilitates seamless and transparent access to HDF5 data. Moreover the system incorporates advanced techniques like concurrent access and large-scale data compression to improve access performance.【Results】After comprehensive technical testing, elaborate performance comparison experiments, and rigorous large-scale data processing evaluations, the XHDF5 system has demonstrated outstanding performance and remarkable stability during remote HDF5 file access. In comparison to traditional access methods that rely on file system mounting and H5serv technology, the XHDF5 system exhibits superior performance particularly in high network latency environments. 【Conclusion】The XHDF5 system effectively supports cross-regional data sharing and collaboration, endowing researchers with a stable and reliable data access platform to facilitate scientific research.

Keywords: XRootD；HDF5；HEPS；NFS；Remote data access

***通信作者：**程耀东（chyd@ihep.ac.cn）

引言

光源大科学装置为我国多学科领域的研究提供了基础研究服务平台。这些装置在科学活动过程中产生了大量具有重要科学研究价值的数据。为满足国家重大战略与基础科学研究需求，高能物理研究所在北京市怀柔区建设一台高性能的第四代同步辐射光源—高能同步辐射光源（High Energy Photon Source, 简称“HEPS”）[1]。HEPS 项目一期将建设 14 条线站和 1 条测试线站，二期将建设 90 多条线站，这些线站预计每天产生的数据量可达 800TB[2]。面对大型科学装置每天产生的海量数据，高效管理和利用这些数据对于科学研究至关重要。

HDF5（Hierarchical Data Format version 5）[3]是一种用于存储和管理科学数据的文件格式，因其效率和灵活性，越来越多地被生物学、天文学和地球科学等领域采用，并被国内光源设施确立为统一数据格式。HDF5 的应用能够有效应对大量科学数据的存储和访问需求[4]。

随着 HEPS 数据中心存储规模的增大和合作成员的增多，“数据集中存储，异地分布处理”的计算模式将被采用，以充分利用异地站点的计算能力。但是，用户在异地站点处理数据时需要提前将数据下载到本地，这是低效的。因此，实现一种高效的、适应性强的 HDF5 数据远程访问方法对于提高科学计算和数据分析的效率和性能至关重要。

本文将探讨如何利用 HDF5 格式及相关技术，实现高效的数据远程访问，从而为 HEPS 及其他大型科学装置的数据处理提供支持。这包括利用 HDF5 库、并行 HDF5[5]以及结合网络传输协议的方法，确保在远程的计算环境下高效处理海量科学数据。通过对这些方法的研究和应用，期望能够显著提升科学数据的分析效率，推动多学科领域的科研进展。

1 背景和动机

1.1 NFS性能分析

目前 HEPS 数据中心对于 HDF5 访问的方式主要就是通过 NFS 挂载的模式访问数据中心的 HDF5 文件。对同一数据中心来说，NFS 协议的同步机制、缓存机制以及高一致性检查，可以有效地保证数据一致性。但是对于远程环境来说，NFS 的同步机制、缓存机制以及请求/响应模型，都会对传输性能带来显著性的影响[6]。

NFS 的同步机制是为了让客户端和服务器在数据一致性和写入操作上进行协调，确保数据写入的正确性和一致性。在同步操作时，每个操作都需要等待服务器进行确认，这会随着网络延迟的增加导致操作变得缓慢。如果每次写入操作的延迟较高，整体性能将受到严重影响[7]。

NFS 基于 RPC 的请求-响应模式，在这种模式下，客户端每发出一个请求，必须等待服务器返回响应才能继续下一个操作。如果采用 mmap[8]方式进行操作，文件进行访问时会出现频繁的缺页异常，每个数据块都需要网络请求并进行确认。 通过对 NFS 的数据报的追踪，发现对于 mmap 的每个缺页操作访问，NFS 都需要发起请求并等待服务器处理每个请求的序列化。在 NFS 和 mmap 的结合使用中，缺页异常的频繁发生意味着需要频繁进行网络请求和响应交互，这种同步的请求-响应模式显著降低了远程文件访问的效率。

在并行访问方面，NFS 支持一定程度的并行访问，但也受到诸多限制。NFS 通常使用单一服务器来处理文件请求，虽然可以通过多线程提高吞吐量，但服务器本身的性能和带宽限制仍会成为瓶颈。并且多个客户端对同一文件进行操作时，NFS 需要频繁进行缓存同步和锁定操作，这在并行访问时可能会降低性能。在高延迟网络下，因为每个请求的往返延迟较大尤其是当客户端与服务器之间需要频繁交互时，并行性无法充分发挥。

1.2 HDF5存储模式分析

HDF5 是一种用于存储和组织大量数据的文件格式。它支持多种类型的数据，并通过分层结构使数据能够以灵活且可扩展的方式存储。HDF5 的底层存储模式通过一组元数据和实际的数据块来实现，其中每个文件包含多个不同的数据结构。

在 HDF5 的存储模式可以有不同地布局方式，主要包括三种模式。第一种是连续存储，数据以线性方式连续存储在磁盘的一个块内，每个数据块的大小是固定的，无法动态调整数据大小。第二种是分块存储，数据分散存储在磁盘内，每个部分可以独立进行管理，因此可以动态改变数据集大小。第三种是紧凑存储模式，数据直接存储在对象头部的元数据信息中，这种方式仅适合存储小规模数据。目前，HEPS 中心 HDF5 文件的存储布局主要采用连续存储，因为分块存储会影响连续读取时的读取效率。

HDF5 在对文件进行读取时，默认会采用 mmap 映射的方式，通过内存映射进行直接将文件内容映

射进入内存中，首先读取数据集的元数据信息，如果是随机读取的话，会通过索引结构获取数据偏移量，得到偏移量之后会直接根据文件偏移量进行数据读取。

1.3 HDF5 VOL

虽然 HDF5 被大量使用，但存储硬件架构和软件系统，如对象数据管理系统(即 Amazon S3[9])已经改变了科学数据管理的方式。为了让 HDF5 用户利用这些新的特性，HDF5 库增加了一个新功能，称为虚拟对象层(Virtual Object Layer，VOL)。VOL 结构允许外部库的开发人员拦截 HDF5 API 并重定向 I/O 调用。

HDF5 VOL 连接器分为终端连接器和直通连接器两种类型，分别负责不同的功能和场景。终端 VOL 连接器拦截 API，然后将数据存储在硬件上，通常采用与 HDF5 不同的文件格式。例如，DAOS[10] VOL 连接器是一种终端 VOL 连接器，它将 HDF5 对象存储为 DAOS 对象格式，直接适配分布式存储系统。直通 VOL 连接器并不直接存储数据，而是将 API 请求传递给另一个 VOL 连接器进行处理。通常，直通 VOL 连接器会与 HDF5 本地 VOL 连接器（或其他终端 VOL 连接器）结合使用，以 HDF5 文件格式存储数据。像异步 I/O VOL 连接器是一个直通 VOL，它使用 HDF5 本地 VOL 连接器作为终端 VOL，以 HDF5 文件格式存储数据[11]。

本文使用的是远程 VOL 连接器，这是一种终端 VOL 连接器，用于实现对 HDF5 文件的远程访问。远程 VOL 连接器通过拦截 HDF5 API 调用，将操作传输到远程服务器，并在服务器端使用 HDF5 本地 VOL 连接器完成对 HDF5 数据的存储与管理。具体细节参考 2.4、2.5 章节。

2 解决方案

针对 NFS 访问方法的局限性和 HDF5 数据存储的特点，以安全快速访问远程数据为目标，本文设计并实现了 XHDF5 系统。在 XHDF5 中采用了异步 I/O、并行访问、数据压缩、Token 认证、权限控制等方法。异步 I/O、并发访问、数据压缩是为了保证远程环境中数据的高性能传输，采用 Token 认证和权限控制方法是为了保证数据中心的数据安全。

2.1 异步I/O

在对 NFS 分析中可以看出 NFS 进行远程传输时频繁的同步操作，在远程环境下，会出现较低的传输性能。在 XHDF5 中采用异步 I/O 为用户提供数据访问接口，保证远程环境下的流畅传输。

在高能物理领域，XRootD 很好地满足了数据密集型应用的需求[12]，得到了广泛应用。它通过高性能和高扩展性的架构，能够处理庞大的数据量，并在多用户环境下保持稳定的并发请求响应能力。XRootD 采用事件驱动模型和多路复用技术实现异步 I/O 机制，通过操作系统提供的高效 I/O 接口（如 epoll），结合内部的任务队列和回调函数，在后台处理文件读写和网络传输等操作。请求发出后，线程不会阻塞，而是将 I/O 操作注册到事件循环中，完成后通过回调机制通知客户端，从而实现高并发、高性能的数据访问与分发能力。

所以，在 XHDF5 系统中采用 XRootD 服务器提供一套 HDF5 文件的异步读写接口，有效避免了类似 NFS 数据传输过程中频繁 I/O 操作所带来的性能瓶颈。同时为了系统的泛用性，在 XRootD 中采用 HTTPs 协议作为用户和数据中心的通信协议[13]。

与 NFS 相比，HTTPs 通过流式传输和更有效的窗口管理，在高延迟条件下能够保持较高的传输效率。并且 HTTPs 协议能够自动加密，具有广泛支持和标准化的证书管理。

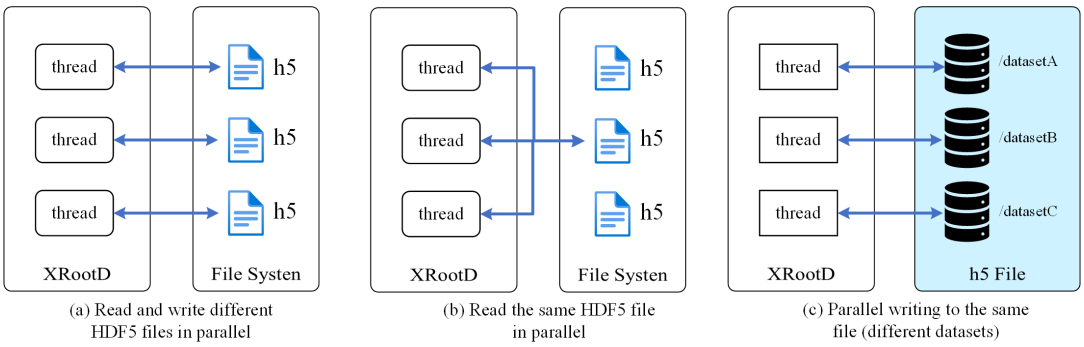

图 1 HDF5 并行访问

Fig. 1. HDF5 parallel operator

2.2 并行访问方法

NFS 在并行访问时受到一定限制，XHDF5 采用 XRootD 作为服务器可以实现并行传输，提高系统的吞吐量，提高系统在远程环境下的性能[14]。

XRootD 部署了多个节点，为用户提供多个数据传送节点。在客户端，用户可以在客户端设置并行传输的服务器节点，VOL 连接器会应用多线程程序并行地从多个服务器节点读取数据。用户可以在环境变量中动态地设置远程服务器的节点数量和节点地址。

对于用户来说，XHDF5 实现了对文件的并发读操作和并发写操作，如图 1 所示。这些并发操作可以在对同一文件进行，也可以对不同文件进行。只是在进行并发写操作时，只能对 HDF5 文件的不同数据集进行操作。

2.3 数据压缩方案

针对大规模数据的传输，XHDF5 会采用无损压缩算法，在服务器端进行数据压缩，客户端拿到数据后再进行解压缩。通过压缩算法可以有效地减少数据传输地耗时。压缩算法的选择主要基于以下几个关键因素：压缩比、压缩的计算开销以及解压缩的计算开销。通过对多种压缩算法的性能进行评估，选择适合的压缩方案，在保证数据完整性的同时，实现传输效率的优化。

2.4 数据安全

首先，系统采用 HTTPs 协议，加密传输数据的网络通信协议，在传输数据时使用 SSL/TLS 协议进行加密，以确保数据在传输过程中的安全性和隐私保护。

对于 HEPS 数据中心来说，需要对用户进行认证，来确定用户的权限，采用 Token 认证的方法[15]，对用户的身份进行认证，同时通过 XRootD 的权限控制列表来为指定的用户或者用户组设置对指定目录的读取、写入、执行等权限。在认证通过后，服务器根据用户的身份和所属组，检查其对文件或目录的访问权限。

在用户的服务器中存在 Token 文件，在 Token 认证时，首先会从用户的本地获取用户的 Token，之后在与服务器通信时传输到服务器中，服务器端会对 Token 进行验证，验证通过会返回用户的用户名。

对于远程用户来说，用户对于远程数据的访问是透明的，其只对某些特定的文件具有访问权限，且只能通过服务器来访问系统的数据。

所以，通过上述用户认证、用户权限管理、数据加密等操作，可以很好保证远程数据安全性。

2.4 系统实现

XHDF5 采用 HTTPS 协议进行数据访问。系统实现如图 2 所示，最上层为用户层，用户通过提供的 API 接口发送数据访问请求。用户请求在 HDF5 VOL 层进行处理，随后通过 HTTP 客户端将用户请求发送到远程服务器。在服务器端，对用户请求进行解析，并进行用户的权限验证，主要采用与高能所系统统一验证方式，通过客户端携带的 Token 代码验证用户的身份。再根据用户请求的信息确定要执行的操作，最终通过 HDF5 本地 I/O 库对底层文件系统中的 HDF5 文件执行相应的操作。

2.5 在应用中使用XHDF5

对于 XHDF5 系统来说，客户端应用了 VOL 框架，堆叠了远程、本地两个连接器，用户只需要设置环境变量 HDF5_PLUGIN_PATH、HDF5_VOL_CONNECTO、SERVER_NUM 和 XRDMULTI_ENDPOINT，其中前两个用来指定连接器的位置，后两个指定服务器的节点数量和 URL。

在 XRootD 中，通过加载共享库地方式为用户提供 read、write 操作的接口，不需要编译整个软件，只需要在配置文件中指定共享库的地址[16]。

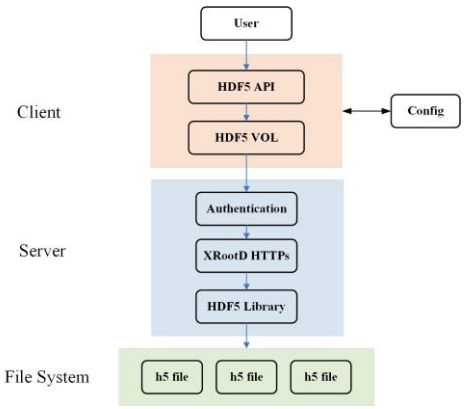

图 2 系统架构

Fig. 2. System architecture

3 实验

3.1 性能评估

在 HEPS 环境中，将 XHDF5 系统与 NFS 进行了对比，在不同的网络延迟下，对性能进行了评估。测试环境的具体结构如图 3 所示。

测试环境采用两台服务器，存储系统使用 NFS

网络协议将底层文件系统挂载到服务器中，即本地服务器(Client1)以及北京怀柔(Server)服务器中，底层文件系统与 Server 在同一数据中心，表 1 中显示了两个服务器的配置。

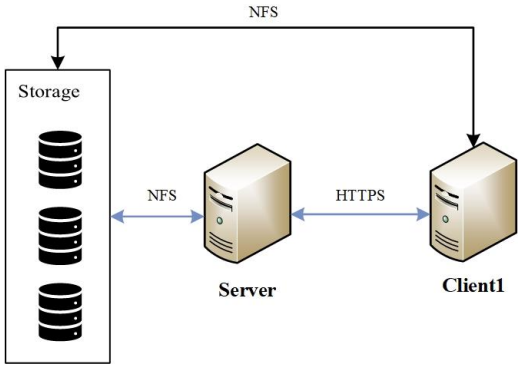

图 3 实验环境

Fig. 3. Experimental environment

存储性能的测试数据集采用光源图像数据集，每张图像大小为 2048×2048 uint16 (8mb)。将 500 张光源图像存储在 HDF5 文件中，在进行测试时会记录 100 张、200 张、300 张、400 张、500 张图像存储的时间。测试的环境以及相关的设备配置如表 1 所示。

在客户端与服务器之间采用 HTTPs 协议进行数据传输，所以采用 iperf 命令对客户端与服务器之间的网络配置与网络带宽进行测试。两个服务器之前的网络带宽约为 10Gbit/s。

表 1 服务器的配置信息

Table 1 The configuration of servers.

	Server	Client1
Memory	503GB	503GB
Mount disk	2.5PB	2.5PB
CPU	112cores	128cores
CPU info	Intel Xeon Gold 6338	Intel Xeon Gold 6330

3.1.1 压缩方案

压缩算法广泛应用于数据存储、传输等领域。随着数据量的急剧增加，选择适合的压缩算法变得尤为重要。ZSTD、ZLIB 和 LZ4 是当前流行的三种压缩算法，它们在速度、压缩率和资源消耗等方面各有优劣。在系统中对这三种算法进行了对比。

测试时，用三种算法压缩 500 张光源图像数据，查看不同算法的压缩比和压缩时间。测试结果如表 2 所示，其中 ZSTD 算法的压缩速度最快，远远超过了 ZLIB 和 LZ4 算法。在压缩比上，ZLIB 的压缩比最高，但是与 ZSTD 算法相差不大，但是压缩速度远远低于 ZSTD 算法。而 LZ4 算法压缩最低，压缩时间仅仅略低于 ZLIB 算法。

表 2 压缩算法对比

Table 2 The configuration of servers.

	压缩比	压缩时间（秒）
ZSTD	1.42	18
ZLIB	1.45	225
LZ4	1.11	180

从测试结果来看，ZSTD 在光源数据集和远程数据传输系统中表现出色。它不仅在压缩速度上领先，还在压缩比上表现良好，尤其适合需要在大规模数据传输中迅速处理的场景。

3.1.2 XHDF5 vs H5serv

H5serv 是基于 REST 的 HDF5 数据存储实现的 Web 服务。提供了一个基于 HTTP 的服务，允许用户通过 Web 浏览器或其他 HTTP 客户端访问 HDF5 文件。

测试时将 H5serv 服务器和基于 XRootD 的服务器都部署在 Server 机器中，在 Client1 上部署对应的客户端。其中，Server 与 Client1 的网络在 1ms 以内。记录读取该数据集中 100 张、200 张、300 张、400 张、500 张的时间。在测试程序中进行重复 50 次上述操作，得到读取的时间。服务器与客户端的默认延迟在 1ms 之内。最后结果以 MB/sec 为单位计算存储速度。最后的结果如图 4 所示，x 轴表示读取文件的大小，y 轴表示文件传输速度，单位为 MB/sec。

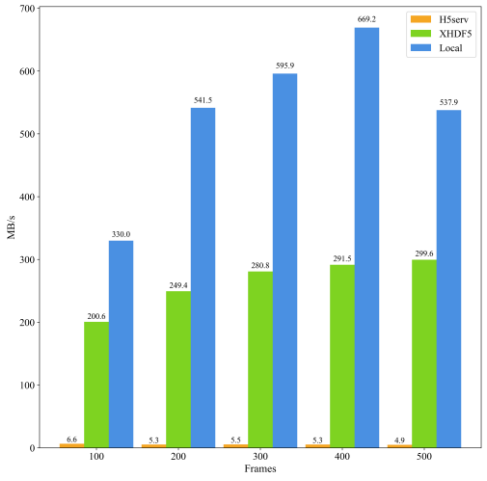

图 4 XHDF5 与 H5serv 的对比结果

Fig 4. Test Results of XHDF5 vs H5serv

橙色的条形图表示使用 H5serv 作为服务器的读取速度，绿色的条形图表示使用 XHDF5 时的速

度，蓝色表示读取本地数据时的速度。从图中可以看出，使用 H5serv 传输速度为 5MB/s，切换 XHDF5 后在 300MB/s 左右。本地 I/O 操作的传输速度会在 380MB/s 到 620MB/s 之间波动。

H5serv 面对大规模文件传输时读取速度较慢是因为 H5serv 采用 json 格式传输数据，而 HDF5 采用二进制存储数据。进行格式转换和数据库存储过程中的 I/O 时间消耗很大，所以其数据访问的速度很慢。对于 XHDF5 来说，在 1ms 延迟的环境中对远程文件系统的访问速度能达到本地文件访问速度的 50%到 75%。

3.1.3 XHDF5 vs NFS

在对系统的性能测试之前首先对 NFS 挂载的文件系统的性能进行测试。针对 NFS 挂载的文件系统，将文件系统挂载到两台服务器（Server 与 Client1）上，采用 IOzone 命令对读、写、混合读写的性能分别进行测试，测试重复 3 次，对测试结果进行求平均，测试了 1 进程、16 进程以及 32 进程下的读写性能，测试结果如图 5、图 6 所示。由于 Server 和 Client1 位于同一个数据中心，因此 Server 上的读写速度高于 Client1 上的速度。在挂载模式下，Client1 实现了 890 MB/s 的读取速度和 1134 MB/s 的写入速度。但是，IOzone 测试仅评估标准文件的读取和写入性能。鉴于 HDF5 文件使用专用的 I/O 库进行读写操作，因此对于 NFS 的性能测试将使 HDF5 的 I/O 库进行。

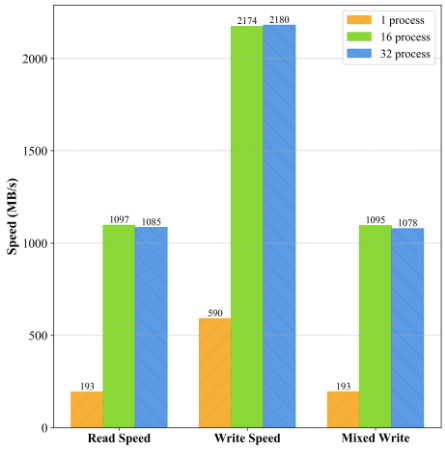

图 5 服务器端 IOzone 测试结果

Fig 5. Server IOzone test result

对 NFS 挂载方式进行对比测试时，将底层文件系统挂载到 Client1 中。在 Client1 中采用 HDf5 的 I/O 库来记录传输文件系中 100 张、200 张、300 张、400 张、500 张 HDF5 文件中光源图片的时间。此外，对于远程用户来说，网络的延迟是一个重要的影响因素，所以在测试时采用 TC 命令在两台服务之前添加网络延时。测试的结果如表 3、表 4 所示。

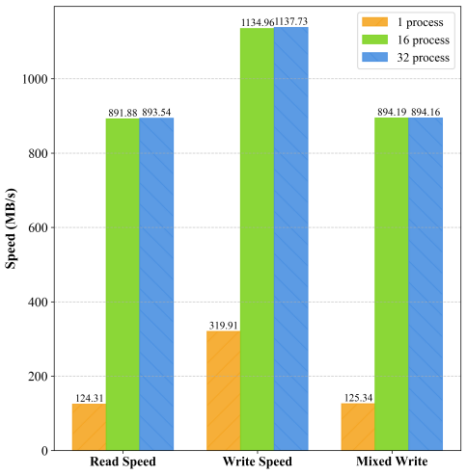

图 6 客户端 IOzone 测试结果

Fig 6. Client IOzone test result

表 3 和表 4 分别列出了在不同网络延迟条件下传输不同大小图片时的传输速度数据。图 7 则展示了传输 500 张图片时，两种方法的传输速度变化情况。折线图清晰地反映出随着网络延迟增加，传输速度的变化趋势。当两台机器之间的网络延迟为 1ms 时，NFS 挂载的效率明显更高，但是当网络延迟超过 5ms 后，XHDF5 系统的传输性能效果明显超过 NFS 挂载。NFS 挂载的方式随着网络延迟的增加，传输速度受到明显的影响，大幅下降。对于远程用户来说，在访问数据时，网络环境的变化显著影响传输效率。特别是，当网络延迟达到 20ms 及以上时，NFS 的传输速度急剧下降，无法满足高性能数据传输的需求。

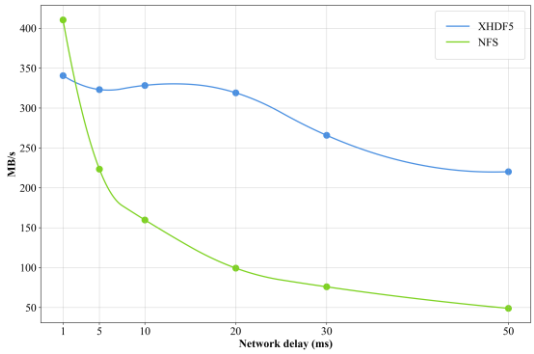

图 7 NFS 与 XHDF5 对比测试

Fig 7. Test Results of NFS vs XHDF5

XHDF5 系统在面对高延迟时表现出了更好的稳定性和传输性能，这使得它在高延迟网络环境下具有明显的优势。XHDF5 系统在 30ms 和 50ms 的延迟情况下，依然保持了较为平稳的传输速度，表明其更适用于远程数据访问和传输。

表 3 采用 XHDF 时的数据传输速度

Table 3 The Data transmission speed of XHDF5.

size\delay	1ms	5ms	10ms	20ms	30ms	50ms
100	204.24 MB/s	240.67 MB/s	219.48 MB/s	211.70 MB/s	182.94 MB/s	144.77 MB/s
200	275.39 MB/s	262.47 MB/s	264.94 MB/s	267.16 MB/s	230.81 MB/s	195.43 MB/s
300	299.44 MB/s	335.34 MB/s	317.84 MB/s	286.43 MB/s	238.24 MB/s	207.68 MB/s
400	287.07 MB/s	321.54 MB/s	346.17 MB/s	300.81 MB/s	267.36 MB/s	225.97 MB/s
500	340.69 MB/s	323.18 MB/s	328.38 MB/s	319.11 MB/s	265.85 MB/s	220.20 MB/s

表 4 采用 NFS 方式的数据传输速度

Table 4 The Data transmission speed of NFS.

size\delay	1ms	5ms	10ms	20ms	30ms	50ms
100	410.5 MB/s	223.4 MB/s	159.8 MB/s	99.4 MB/s	76 MB/s	49 MB/s
200	429.4 MB/s	228.7 MB/s	169 MB/s	112.4 MB/s	79.3 MB/s	50.9 MB/s
300	504.6 MB/s	232.9 MB/s	168.3 MB/s	111.5 MB/s	78.9 MB/s	51.9 MB/s
400	412.7 MB/s	246.6 MB/s	169.8 MB/s	109.4 MB/s	80.3 MB/s	52 MB/s
500	462.4 MB/s	252.9 MB/s	172.5 MB/s	112.4 MB/s	80.3 MB/s	52.4 MB/s

具体来看，在 1ms 延迟下，XHDF5 的传输速度为 340 MB/s，而 NFS 则达到 462.4 MB/s。随着延迟的增加，NFS 的传输速度逐渐下降，例如在 5ms 延迟时降至 252.9 MB/s，而 XHDF5 为 323MB/s。进一步增加延迟至 10ms 和 20ms 时，NFS 的传输速度分别降至 172.5 MB/s 和 112.4 MB/s，而 XHDF5 则分别为 328 MB/s 和 319 MB/s，在 20ms 的延迟之内，XHDF5 速度并没有明显下降。最终在 50ms 延迟时，NFS 的传输速度降至 52.4 MB/s，而 XHDF5 依然保持在 220 MB/s，可以满足正常的访问需求。

此外，为了评估在多线程环境下，系统在不同网络延时下的性能表现，我们设置了客户端对服务器的并发访问实验。具体地，客户端以不同的线程并发数（分别为 1、16、32）进行服务器访问，每个线程负责访问 500 张光源图像数据。通过这些实验，我们能够观察到系统在不同并发线程数和网络延时下的整体传输性能。实验结果如图 8 所示，展示了随着并发线程数和延时的变化，系统性能的不同表现。

从图 8 中可以看出当并发进程达到 16 进程时，访问速度达到了将近 680MB/s，32 进程时仅有略微提升，说明在 16 进程时机器已经达到了满带宽，之后添加延时，系统的总体传输速度并未收到太大影响，基本在同一水平线内，在 50ms 的延时下，系统依然能达到将近 600MB/s 的传输速度。

这些数据清楚地表明，NFS 在低延迟情况下表现优异，但其性能随着网络延迟的增加迅速下降。而 XHDF5 则在高延迟情况下表现更为稳定，这对于需要跨地域或在不稳定网络环境中进行远程数据传输的用户来说尤为重要。对于科研数据的远程传输，选择合适的传输协议和系统至关重要。在低延迟的局域网环境中，NFS 或许是一个不错的选择，但在广域网或者跨地域的数据传输场景中，XHDF5 无疑更具优势。

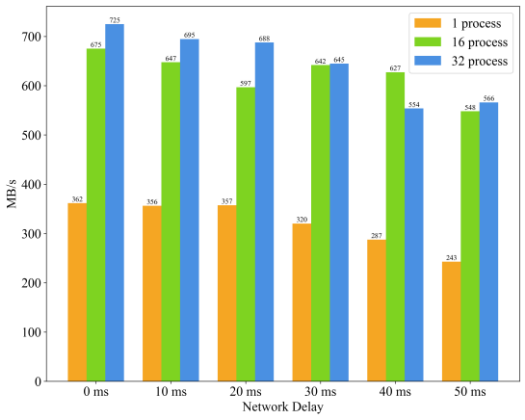

图 8 多线程时的 XHDF5 的传输速度

Fig 8. XHDF5 transfer speed when multithreading

通过以上分析可以得出，XHDF5 系统在高延迟网络环境下的优势，使其成为远程数据传输的优选方案。而 NFS 则更适合在 HEPS 数据中心内部使用。

3.2 压力测试

为了更全面地评估系统在高并发条件下的性

能，设计了多客户端并发测试方案。在测试中，配置了三个客户端，每个客户端同时发起 32 个请求，每个请求都会向服务器请求 4GB 的数据。

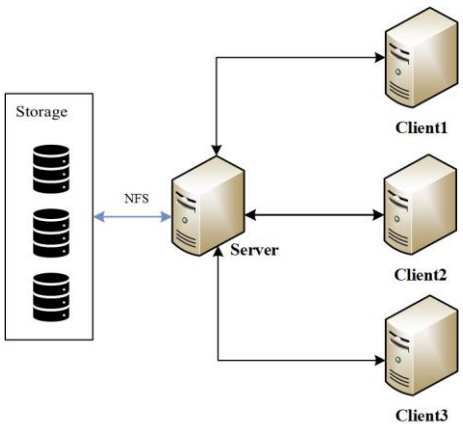

图 9 压力测试的环境

Fig 9. Test Environment of Stress Test

在这种测试场景下，重点关注以下几个方面的性能指标：

(1) 网络吞吐量：测量每秒钟传输的数据量，以评估网络是否能够支持如此大规模的数据传输。观察整体网络的利用率，也就是服务器端的聚合带宽。

(2) 系统资源消耗：监控服务器的 CPU 和内存的使用情况，特别是当多个客户端同时请求大量数据时，观察系统是否出现资源耗尽或性能下降的现象。

(3) 稳定性：评估系统在长时间高并发下的运行稳定性，记录是否出现请求失败、超时或其他错误情况，以及这些问题的频率和原因。

因此，设计了以下实验：

(1) 将 HDF5 客户端部署在 3 台机器中，将基于 XRootD 的服务器部署在 Server 中，机器配置与表 1 一样，机器的部署如图 9 示。

(2) 每个客户端的并发线程设置为 32，每个线程内部向服务器访问的数据大小为 4GB。

(3) 设置不同的网络延迟（1ms、10ms、20ms、30ms、40ms、50ms），在相同并发量和数据大小条件下，记录整体的聚合性能以及资源占用情况。

从图 10 中可以看出，系统的整体网络带宽在不同的网络延时下表现出较高的稳定性。即使网络延迟有所增加，系统的最大网络带宽仅略有下降。这表明系统的网络传输性能在面对延迟变化时具有良好的适应性。系统的平均网络带宽维持在 2.7GB/s，并达到了系统网络带宽的极限。这表明系统的网络架构和硬件设计能够在较高负载下保持稳定的性能表现，没有出现显著的性能瓶颈。在高网络带宽下，即使面对增加的网络延迟，系统仍然能够平稳运行，且没有出现错误。这进一步说明系统在大规模数据传输场景下的稳定性和可靠性。即使在网络延迟增大的情况下，系统依然能够保持接近极限的网络带宽，且运行稳定，这对于处理大规模数据的应用场景非常关键。

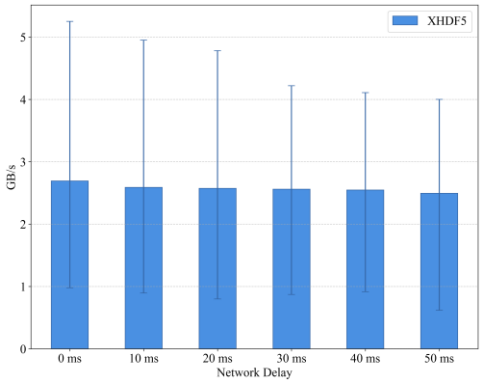

图 10 压力测试的系统带宽

Fig 10. Stress test system bandwidth

此外，也对存储程序运行期间的 CPU 和内存利用率进行了监控，结果如图 11 所示。图中，x 轴表示时间（单位为秒），y 轴表示利用率（单位为百分比）。其中，蓝色线表示系统的内存利用率，黄色线表示缓存利用率，绿色线表示 CPU 利用率。

对于服务器端而言，当存储程序运行时，CPU 利用率会显著增加，结束后则有所回落，但整个过程中 CPU 利用率从未超过 65%。内存利用率的趋势与 CPU 类似，最高也维持在 65%左右，表明内存负载同样不是系统的瓶颈。

对于三个客户端而言，CPU 利用率最高仅为 10%，而内存利用率最高不超过 7.5%。因此，在客户端和服务器端，CPU 和内存的负载都不是限制系统吞吐量的关键因素。

基于以上观察，可以推断，对传输性能影响较大的可能是 HDF5 文件的 I/O 性能。这与 HDF5 的 I/O 读写库的效率密切相关，因此，优化 HDF5 I/O 操作将有助于提升整体存储和传输性能。

3.3 总结

在 XHDF5 系统的性能测试中，对比了 H5serv 和 NFS 这两种远程访问方式，其中 XHDF5 性能远

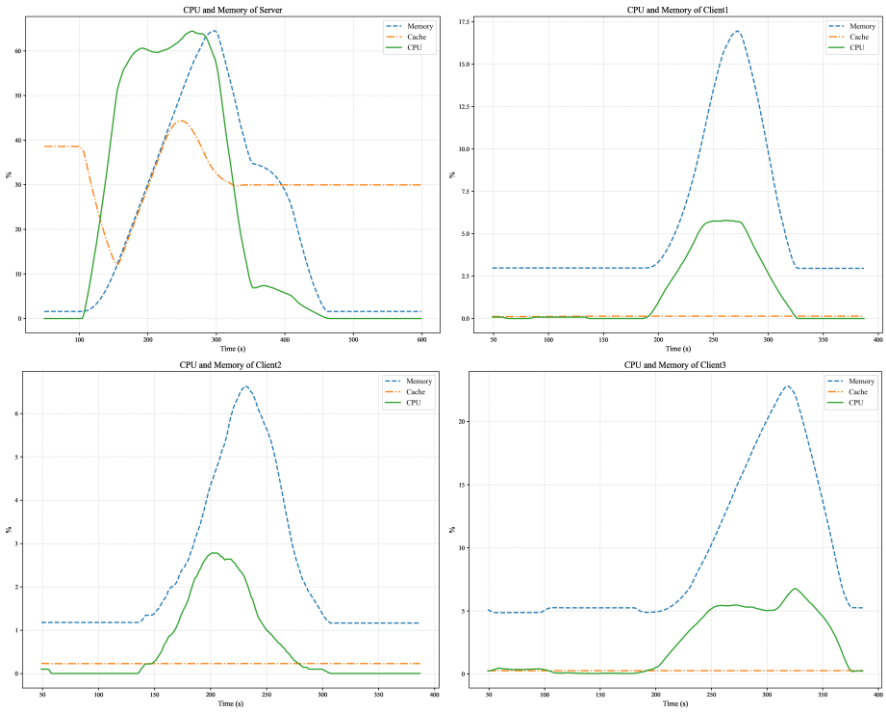

图 11 CPU 和内存占用

Fig 11 CPU and Memory utilization

远高于 H5serv，在低延迟（延迟小于 1ms）时，NFS 方式进行数据访问的性能较好，但是当延迟高于 5ms，XHDF5 的性能就已经超过了 NFS 方式，在超过 50ms 时，NFS 性能远远低于 XHDF5，所以在实际的远程环境中，XHDF5 的性能要远远优于其他两种方法。

之后还对系统进行了压力测试，测试了在上百个进程同时对单个 Server 进行大规模的数据访问时系统的性能，实验结果表明，在高性能模式，Server 依然能保持极限的带宽输出，并且系统运行稳定，这说明系统也适用于大规模数据访问的场景。

4 结论

高能同步辐射光源每年产生的数据量高达 300PB，这些数据需要进行复杂的数据分析。为了显著提高用户的分析效率，本文基于 XRootD 和 HDF5 VOL 技术，设计并实现了 XHDF5 系统，提供了对 HDF5 数据的高性能远程访问。XHDF5 系统支持对服务器的动态扩展，能够灵活地满足用户不断变化的访问需求在性能测试中，对比了 H5serv 和 NFS 方法，与这个两个访问手段相比，XHDF5 的优势也很显著：（1）在较高网络延迟的环境下，具有更好的性能；（2）提供透明的访问接口，不用进行较大代码改动；（3）提供灵活的权限控制机制和完善的用户认证策略来保证数据的安全性。在压力测试中，设置了上百个用户，在进行高达上百 GB 数据的实时访问时，系统表现了良好的性能和稳定性。

综上所述，XHDF5 系统可以满足远程用户对数据中心 HDF5 文件的访问需求，在保证数据安全性的同时，能极大提高用户对科学数据的分析效率。

致谢

本项目由中国科学院高能物理研究所谢家麟基金项目支持(E35464U2)。

利益冲突声明

所有作者声明不存在利益冲突关系。

收稿日期：2024 年 11 月 21 日

参考文献

[1] HU Q, XU J, CHENG Y, et al. Design and implementation of experimental data access security policy for HEPS container computing platform[C]. EPJ Web of Conferences. EDP Sciences, 2024.

[2] 齐法制, 黄秋兰, 胡皓, 等.高能同步辐射光源科学数据处理平台规划与设计[J]. 数据与计算发展前沿,2020,2(02):40-58.

[3] BYNA S, BREITENFELD M S, DONG Bin, et al. ExaHDF5:Delivering efficient parallel I/O on

exascale computing systems[J]. Journal of Computer Science and Technology,2020,35(1):145-160.
[4] 严飞，瞿铸枫，张立强.面向 HDF5 格式预训练模型的模糊测试方法[J].郑州大学学报(理学版),2023,55(1):1-7.
[5] LEE S, HOU K, WANG K, et al. A case study on parallel HDF5 dataset concatenation for high energy physics data analysis. Parallel Computing, 2022, 110: 102877.
[6] PEYROUZE N, MULLER G. FT-NFS: An efficient fault-tolerant NFS server designed for off-the-shelf workstations[C]. Proceedings of Annual Symposium on Fault Tolerant Computing. IEEE, 1996:64-73.
[7] CHENG X, LIANG Q, HE J. Research on objective weight determine method of ZigBee network performance index[J]. Application Research of Computers/Jisuanji Yingyong Yanjiu, 2013, 30.10.
[8] VAN ESSEN B, HSIEH H, AMES S, et al. DI-MMAP—a scalable memory-map runtime for out-of-core data-intensive applications[J]. Cluster Computing, 2015, 18: 15-28.
[9] GADBAN F, KUNKEL J. Analyzing the Performance of the S3 Object Storage API for HPC Workloads[C]. Applied Sciences, 2021, 11.18: 8540.
[10] HENNECKE M. Daos: A scale-out high performance storage stack for storage class memory[C]. Supercomputing frontiers, 2020, 40.
[11] ZHENG H, VISHWANATH V, KOZIOL Q, et al. HDF5 Cache VOL: Efficient and scalable parallel I/O through caching data on node-local storage[C]. 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid). IEEE, 2022:61-70.
[12] DORIGO A, ELMER P, FURANO F, et al. XROOTD-A Highly scalable architecture for data access[J]. WSEAS Transactions on Computers, 2005, 1.4.3: 348-353.
[13] FAJARDO E, ARORA A, DAVILA D, et al. Systematic benchmarking of HTTPS third party copy on 100Gbps links using XRootD[C]. EPJ Web of Conferences. EDP Sciences, 2021:02001.
[14] 杨丽鹏,车永刚.基于 HDF5 实现多区结构网格 CFD 程序的并行 I/O[J].计算机研究与发展,2015,52(04):861-868.
[15] KHAN H, ZAHID H. Comparative study of authentication techniques[J]. International Journal of Video & Image Processing and Network Security IJVIPNS, 2010, 10.04: 09-13.
[16] IORDACHE C ,LIU R ,BALCAS J , et al.Named Data Networking based File Access for XRootD[J].EPJ Web of Conferences,2020,245 04018-.

程耀东，中国科学院高能物理研究所，研究员，博士生导师，主要研究领域为云计算、海量存储和大数据。
本文承担工作为：研究指导，整体框架的设计规划，论文的审定。

CHENG Yaodong is a professor at the Institute of High Energy Physics, Chinese Academy of Sciences. His research interests include cloud computing, mass storage and big data.
The work of this paper is as follows: research guidance, design and planning of the overall framework, and examination and approval of the paper.
Email: chyd@ihep.ac.cn

冯时超，郑州大学计算机与人工智能学院，硕士研究生，主要研究方向是 HDF5 远程数据访问方法及应用研究。
本文承担工作：XHDF5 系统的设计与实现，论文的初稿撰写。

FENG Shichao is a master student at the School of Computer and Artificial Intelligence, Zhengzhou University. His research interests are HDF5 remote Data access methods and applications.
The work undertaken in this paper includes: the paper drafting, and XHDF5 development.
Email: scfeng@ihep.ac.cn

程垚松，中国科学院高能物理研究所，工程师，主要研究领域为网络运维及存储研发。
本文承担工作：研究指导，论文的修改、审定。

CHENG Yaosong is an engineer at the Institute of High Energy Physics, Chinese Academy of Sciences. His research interests include network operation and maintenance and storage research.
The responsibilities of this paper: research guidance, revision and approval of the paper.
Email: chengys@ihep.ac.cn