基于NFS协议
构建高性能并行文件系统

技术综述

综述

业界对于高性能运算（HPC）的需求突然变得高涨
传统上，“HPC”通常指的是那些罕见的学术研究环境以及保密级别的政府实验室。然而，随着近年来AI及其相关工具的迅猛发展，大型计算集群不仅开始在各类企业数据中心内出现，也越来越多地被部署于公有云平台之上。HPC技术走向大众化的过程对数据存储提出了新的要求——存储系统不仅要能处理更大规模的数据，还要具备更高的效率。现在，数据不仅要在多个地理位置之间进行创建、存储和处理，甚至可能跨越不同的云端环境，这就要求数据具有高度的流动性。

Hammerspace长期以来一直在与国际互联网工程任务组（IETF）和Linux社区紧密合作，致力于研发基于标准协议的开源解决方案，以应对上述挑战。这些研究成果体现在NFS v4.2和pNFS标准中，这两项技术是支撑Hammerspace超大规模NAS架构的关键。

通过采用NFS v4.2和pNFS标准，用户能够利用通用硬件和软件资源搭建起既高效又易于扩展的并行文件系统。此外，这些标准还引入了一些特别设计的功能，用以支持数据的无缝迁移，而这正是Hammerspace所擅长的领域之一。本文档将详细介绍NFS v4.2如何在Hammerspace全球数据环境中发挥其独特的优势。

性能也需要并行化

如果说CPU和GPU是高性能计算引擎，那么数据就是驱动引擎的燃料。如同电动机需要稳定的电流供应，计算集群必须保持数据流的持续畅通才能释放全部潜能。无论是计算还是存储，并行化架构都是实现高吞吐、可扩展性的核心路径。

并行文件系统的核心设计哲学在于元数据与数据分离——元数据：处理零散随机的小型I/O请求；数据流：承载连续有序的大规模传输。这种解耦架构能够带来多重优势，最关键的是允许客户端直连存储设备，彻底摆脱传统NAS节点可能造成的性能瓶颈。

目录

- HPC无处不在，数据存储架构需与时俱进
- 并行NFS（pNFS）发展史简介
- Hammerspace如何通过pNFS构建基于标准协议的并行文件系统
- 实际应用：超大规模AI训练环境的成功案例

HAMMERSPACE

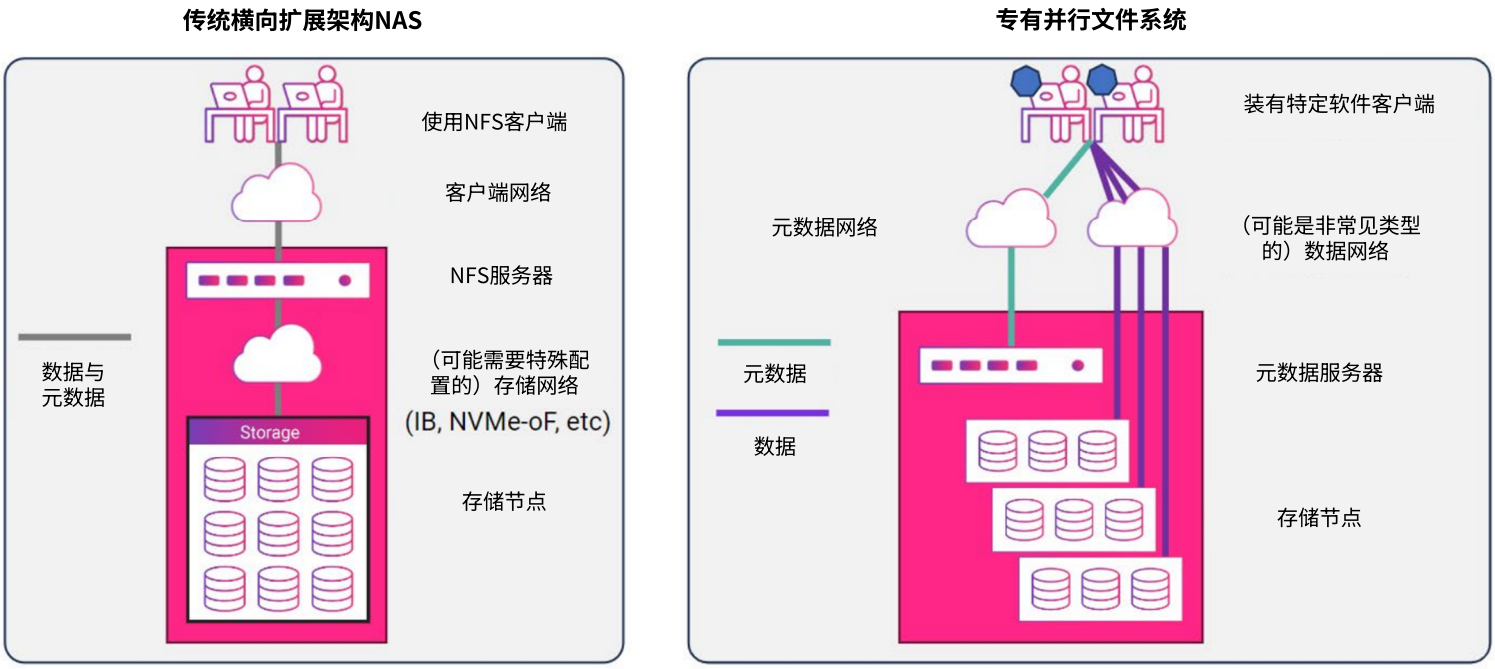


Figure 1 -传统横向扩展NAS与传统专有并行文件系统对比

在并行架构中，客户端访问数据时首先与元数据服务器通信，获取存储路径映射和访问凭证。随后，客户端即可直接读写存储设备，数据路径中不再存在任何中间层。在过去，系统的规模和性能受限于可以直接挂载在阵列控制器或NAS前端的存储设备数量；而如今，并行架构允许客户端将文件条带化分布到多个存储系统中，这种方式不仅提升了性能，也增加了可用的存储容量。

过去，传统专有并行文件系统长期受限于封闭生态：比如，每台访问设备必须安装专用客户端软件；客户端与操作系统版本深度绑定，灵活性受限；授权费用与客户端数量/类型挂钩，大规模部署成本高昂。这些限制在小规模环境中尚可接受，但对于拥有数千甚至数万节点的集群而言，已成为重大瓶颈。而**NFS协议的革新**彻底打破了这些传统架构的桎梏。

关于并行文件系统（Parallel NFS/pNFS）的一些简短历史回顾

NFSv3 如何脱颖而出

NFSv3凭借数十年的成功实践，已成为客户端访问网络存储的行业标准，并获得了存储和操作系统厂商的全面支持。然而，随着时间推移，NFSv3的局限性逐渐显现：由于采用无状态连接模型，单客户端挂载单NFS导致的性能存在瓶颈，尽管业界开发了多种性能优化方案，但这些方案均未被纳入标准。

而NFSv4于2000年首次发布，并在后续多次更新。相比NFSv3，最显著的变革是从无状态协议演进为有状态协议。这一转变虽然增加了协议复杂度，但为功能扩展提供了基础框架。但这样演进并非一帆风顺：早期NFSv4因协议交互频繁、性能表现不佳而饱受批评；这些质疑在当时确实是有依据的，但后续版本已彻底解决这些问题。当前版本不仅在传输效率上大幅优化，性能也全面超越了v3和早期的v4版本。

并行文件系统（Parallel NFS/pNFS）的出现

由无状态转变为有状态所带来的新增功能之一是并行 NFS（pNFS），该功能在2010年作为NFSv4.1的一个可选特性被引入，并在后续的RFC标准中得到了增强。

pNFS定义了一种NFS架构，类似于前面提到的专有高性能文件系统。在这种架构中，元数据和数据路径是分开的。一旦元数据服务器授予访问权限，客户端就可以直接与存储进行通信，从而实现并行读取或写入数据的能力。

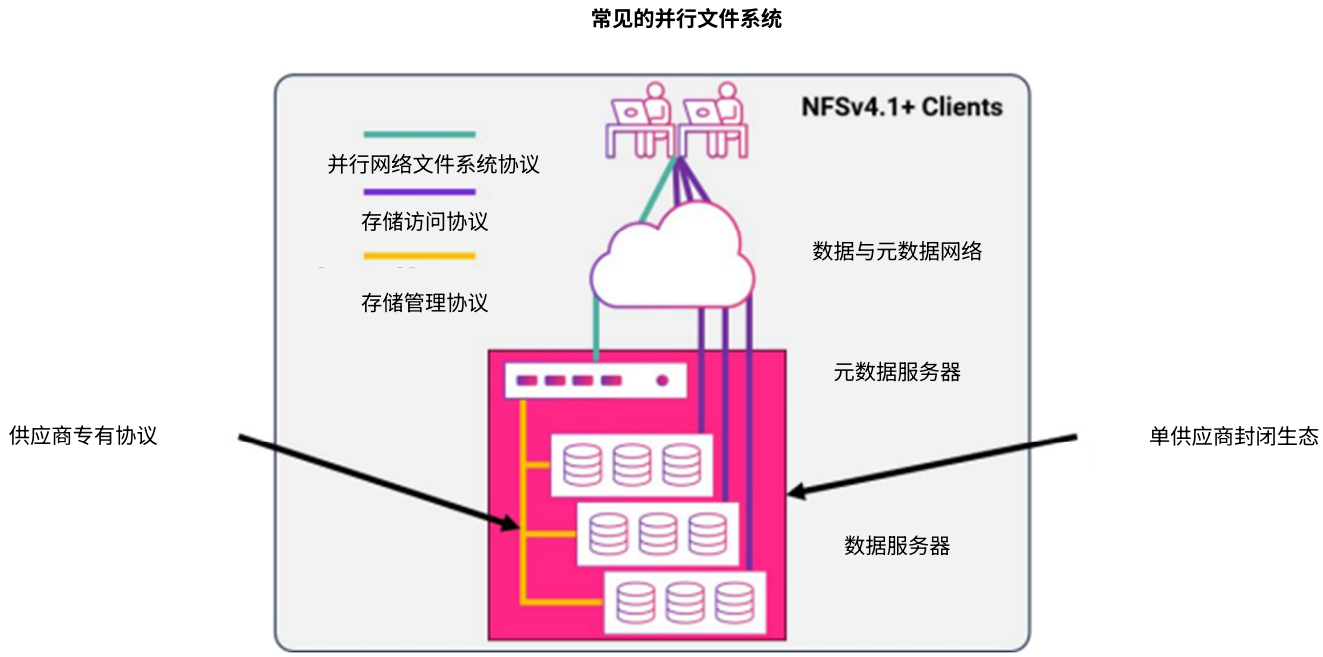


Figure 2 - Parallel NFS 架构

在这一新架构中，元数据服务器与数据服务器的角色相互分离。系统定义了三种协议用于服务器与客户端之间的通信：

- **pNFS协议**：用于客户端与元数据服务器之间的元数据路径通信
- **存储访问协议**：用于数据路径通信，可以是 NFSv4.1 或其他多种协议。
  - 额外的 RFC（称为布局规范）定义了 pNFS 如何使用块/卷存储协议(RFC 5663)和对象存储协议(RFC 5664)
- **存储管理协议：**用于元数据服务器与数据服务器之间的通信，以处理存储分配与释放、客户端访问控制所需的状态管理及其他功能

需要注意的是，虽然定义了存储管理协议的角色，但该协议的具体细节并没有明确规定。这部分内容留给了各个厂商自行实现，比如元数据服务器和存储服务器可以是存储厂商NAS产品中的互补组件。因此，pNFS的服务器端实际上是专有的。所以，尽管它比非并行NFS有所改进，但仍然不是最理想的解决方案。

为提升性能和扩展性而增加的额外功能包括：

- 支持使用nconnect选项（NFSv3的可选功能）在客户端与数据服务器之间建立多条并行网络连接
- 可选使用 RDMA（远程直接内存访问）
- 数据（即使是单个文件）也可以跨多个数据服务器进行条带化或镜像化存储。这使得即使单个数据服务器的 可靠性和性能较低，也能构建出具有高性能、高可靠性和高可用性的系统
- 能够在客户端访问数据的同时无缝迁移数据，而不会中断I/O操作

要使用这一新架构，兼容的NFS客户端在访问数据时首先会联系元数据服务器。元数据服务器提供一个布局（layout），其中包含字节范围和存储位置信息。当客户端持有该布局时，可以直接访问相应的存储位置。布局是一种抽象机制，使客户端无需了解底层存储的具体细节。

FlexFiles让可能变为现实

Hammerspace看到了一个机会，可以让pNFS更加开放并增加重要的新功能。我们在2018年定义了灵活文件（Flexible Files，简称FlexFiles）布局类型，如RFC 8435所述。在FlexFiles布局中，使用 NFSv3 或更高版本作为存储协议，同时也作为存储管理协议。

这意味着，现在用户可以使用pNFS元数据服务器，加上任意组合的NFS存储服务器来创建高性能并行NAS系统。这些存储服务器可以来自任何厂商、混合厂商，或者由通用硬件自定义构建，只要它们支持 NFSv3 即可。由于存储管理协议的功能也通过NFS定义，因此在使用FlexFiles时，pNFS不再具有任何专有特性。

带FlexFiles布局的并行NFS

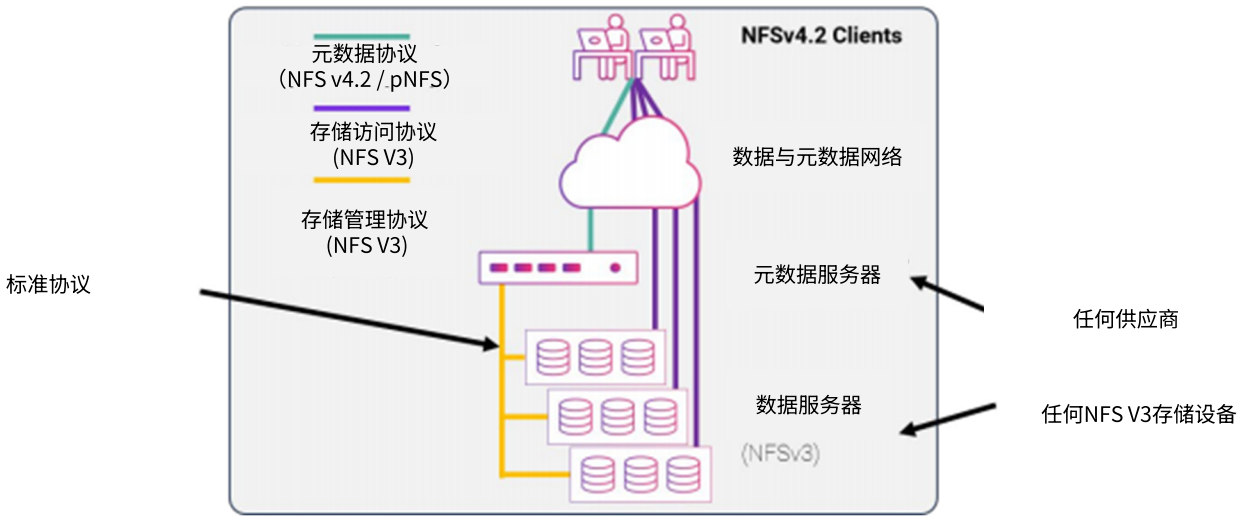


Figure 3 - pNFS with FlexFiles

作为额外的功能，FlexFiles 定义还包括了客户端镜像（client-side mirroring）的规范。在该机制中，元数据服务器会向客户端共享所有镜像的位置信息。随后，客户端可以选择一个镜像进行读取操作，并在执行写入操作时负责更新所有副本。这一设计将镜像功能从单个存储设备中剥离出来，使其能够随客户端数量的增加而扩展。

NFSv4.2 定义了更丰富的特性

Hammerspace与国际互联网工程任务组（IETF）共同制定了NFSv4.2协议规范，该规范在NFSv4.1基础上定义了一系列可选扩展功能。正如RFC文档所述：NFSv4.2的核心目标是将传统本地文件系统的特性引入远程文件访问场景，弥补早期NFS版本的功能缺失。其关键技术特性包括：

- **服务器端克隆和复制功能**允许客户端请求存储服务器执行同步克隆以及服务器内部和跨服务器文件复制（后者可以是异步的）；这减少了必须通过网络传输的数据量：从原先需要传输两倍数据（先传给客户端再传回）减少到仅需传输一次（对于存储服务器之间的复制），而对于单个存储服务器内的复制和克隆则无需传输任何数据。

- **应用程序 I/O 建议机制**提供了一种方法，让客户端能够告知存储服务器关于预期未来的 I/O 行为，例如文件是否将被顺序或随机访问，以及文件是否会在不久后被访问。这使得服务器能够做出智能的存储分配决策，以优化未来的访问性能。

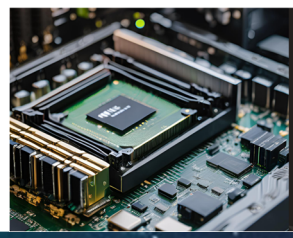
- **稀疏文件和支持空间预留功能**使存储服务器能够高效地使用元数据通知客户端文件中的“空洞”，而不是简单地传输零值数据。这显著地降低了网络流量。分配函数允许客户端请求保证文件的空间可用性，相应的释放函数允许客户端在文件中创建一个空洞，释放预留空间，优化存储资源利用率。

- **应用程序数据块支持功能**提供WRITE_SAME操作，支持单次元数据传输完成文件初始化；减少类磁盘操作场景下的网络负载。

- **标签化 NFS** 实现了一个新的文件对象属性 sec_label，使存储服务器能够在文件上存储强制访问控制（MAC）标签，以便客户端能够在支持 MAC 的环境中实施数据访问限制，实现细粒度的数据访问控制。

- **布局增强**弥补了错误可见性和性能方面的差距。在 pNFS 之前，数据访问是通过单一服务器或“NAS 机头”进行的，它们自然可以看到底层存储上的错误和性能行为。将元数据服务器功能与数据路径分离后，元数据服务器无法看到客户端与存储交互时的体验。

- 而NFSv4.2 包含两种新的操作行为，用于客户端将他们的体验反馈给元数据服务器：**LAYOUTERROR** 允许报告存储目标上发生的持久性错误，提供了元数据服务器调整和/或报告问题的机会。**LAYOUTSTATS** 允许客户端定期报告他们对特定存储目标的性能体验。这使得元数据服务器能够基于这些信息进行报告和采取行动。



如需了解更多-可联系
ralphie.yu@hammerspace.asia
或 18810311799
手机・微信同号
HAMMERSPACE

不要忽视客户端的重要性

这些扩展增强功能都非常出色，但要充分利用它们不仅需要元数据服务器，还需要理解 pNFS 和 NFSv4.2 的客户端。幸运的是，在 pNFS 可以发挥最大优势的大规模应用场景中，只有一个客户端是最重要的——那就是 Linux NFS 客户端。

Hammerspace 也在这一领域进行了不懈的努力。Hammerspace 的首席技术官 Trond Myklebust 担任 Linux 内核 NFS 客户端维护者已经超过二十年。他与社区合作，始终确保新出现的 NFS特性和功能可以在 Linux 上得到稳健的支持。例如，Red Hat Enterprise Linux (RHEL) 自 2017 年发布的 7.4 版本起，就已经包含了对 NFSv4.2 的支持。

Hammerspace 与 Parallel NFS

Hammerspace 架构概览

Hammerspace 的全局数据环境通过高性能并行全局文件系统提供统一的文件访问，该文件系统可以跨越任何厂商的任何存储设备类型以及地理位置及云环境。如下所示，您可以对Hammerspace 的架构有一个大体的了解。

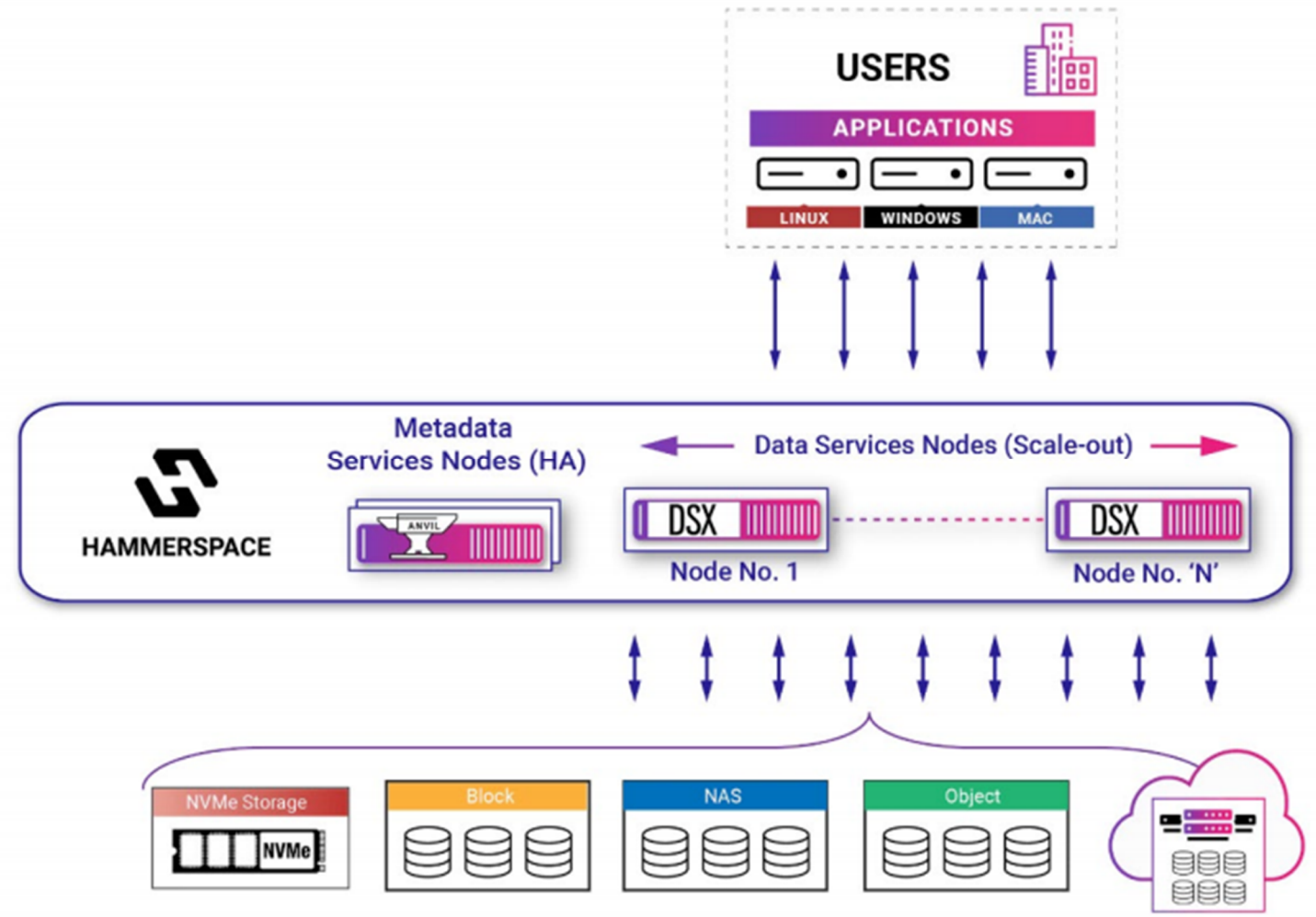


Figure 4 - Hammerspace 架构

Hammerspace软件采用横向扩展架构部署，其集群由两种协同工作的节点类型组成，作为一个统一系统运行：Anvil节点和DSX节点。

- **Anvil 元数据服务节点**（通常以HA Pair的形式部署）承载元数据控制平面并驱动系统的智能化。文件 I/O 不会通过 Anvil 节点进行。

- **DSX 数据服务节点**处理 I/O 操作，包括数据编排服务，并设计为可横向扩展以满足任何性能需求。它们可以承担以下三个功能中的一个或多个:

  - **Store:** 当块存储（如 NVMe 闪存驱动器）连接到 DSX 节点时，该节点可以将其作为并行全局文件系统的一部分实现共享。

  - **Mover:** 执行所有数据迁移操作，例如基于策略目标的数据复制或迁移

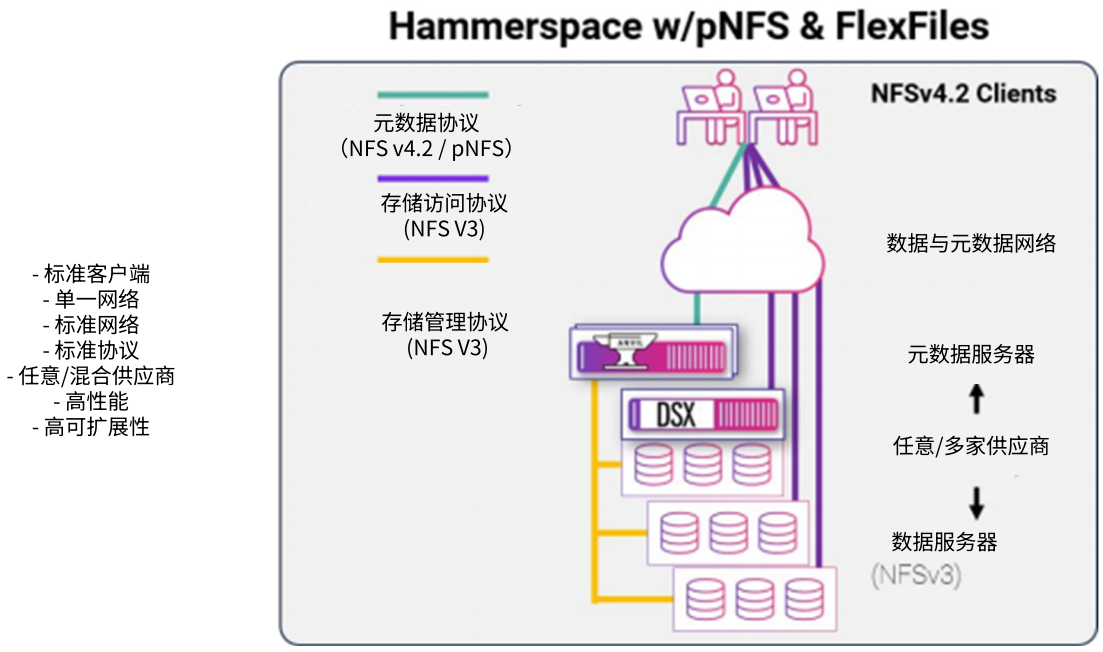
  - **Portal:** 为不支持 NFSv4.2 的传统客户端提供支持，使得 Hammerspace 文件系统可以通过 SMB、CSI 和旧版本的 NFS 进行呈现

除了由Hammerspace DSX节点直接提供的存储外，我们还可整合来自任何厂商的NAS、对象存储、云存储和磁带存储。Anvil节点会同步这些存储系统的元数据，使Hammerspace能够在全局数据环境中对它们进行管理和使用。

如需了解Hammerspace架构和功能的更多详细信息，请参阅《Hammerspace技术白皮书》。

Parallel NFS with Hammerspace

Hammerspace 始终期望能够充分利用 NFSv4.2 和 pNFS的特性，使客户能够基于通用组件，创建具备 HPC 级性能的标准并行文件系统。下图将前面的 pNFS 和 Hammerspace 架构图层结合在了一起，以展示其的工作原理：



在这种配置中，Anvil 节点充当 pNFS 元数据节点，而数据服务器可以是 DSX 节点——或者是带有附加块存储功能的普通 Linux 服务器。客户端与 Anvil 节点之间的元数据通信通过 NFSv4.2 进行，客户端与数据服务器之间的数据通信则使用 NFSv3。由于 Hammerspace 使用 FlexFiles 布局类型，Anvil 和 DSX 节点之间使用的存储管理协议也是 NFSv3。

请注意，当使用带有块存储功能的 Linux 服务器作为数据服务器时，NFS 客户端软件和服务器软件都是标准的开源组件；而且因为 Hammerspace 的 Anvil 节点运行在 Linux 上，也不需要用户另行采购或使用专有硬件。

这样的开放性是使 Hammerspace 在大规模应用环境中极具吸引力的特点之一。管理成千上万台 Linux 服务器的组织，通常由自己的自定义内核而构建。那些需要加载自定义客户端软件或仅限于特定发行版、构建版本或内核版本的解决方案往往并不适用。根据具体情况，远程直接内存访问（RDMA）以及所需特殊网络（InfiniBand或无损以太网）可能受到冷遇，也可能是必需的。

Proof Point: 超大规模AI训练环境

提现超大规模NAS架构的强大之处的一个很好的例子，就是它在Hammerspace的一位客户的大规模语言模型（LLM）训练环境中得到的应用。

该客户的配置包括800台Linux存储节点作为数据服务器，提供了总计29PB的NVMe存储。计算集群则由2,000台Linux服务器作为文件系统客户端组成，总共搭载了16,000个GPU。在这个配置下，系统实现了7.2TB/s的实际吞吐量，提供了真正的“HPC级别”的性能，而在过去，这种性能水平只能通过专有系统才能达到。

有关此案例实现的更多详情，请参阅Hammerspace AI解决方案页面上的新闻稿。

结论

得益于Hammerspace公司、国际互联网工程任务组（IETF）以及Linux社区的努力，如今利用标准硬件和软件组件构建高性能并行文件系统已成为可能——无需专有客户端、非标准配置网络或特殊硬件。

Hammerspace在其超大规模NAS架构中充分利用了网络文件系统（NFS）的发展成果。成为了一款完全基于标准的高性能并行文件系统，具备企业级的易用性和全局数据编排能力。

