白皮书

CH StorNext 横向扩展文件系统

架构、特征和优势

目录

简介

世界上被创建并管理的数据量正在以惊人的速度进行增长。根据 IDC 的统计，2018 年“全球数据规模”达到了 33 Zettabytes（ZB），预计到 2025 年将达到 175 ZB。这些数据中很大一部分是以视频、图像和类似的非结构化内容形式存在的。造成这一数据爆炸的原因来源于数据采集设备-从手机摄像头到卫星传感器-这些终端正在持续不断地以更高的分辨率、更高的保真度和更高的采样率产出上述内容。我们可预见到的是，数据不仅在持续增长，相关的文件大小和文件流也变得越来越大，并且需要更高性能的系统对其进行存储和使用。

存储越来越大的文件，越来越快的生成和传输速度，一直是数据存储系统面临的挑战。StorNext®在 20 多年前创建时，正是为了应对这一挑战，特别是生成、管理和共享高分辨率数字视频，以及拥有其他高性能要求文件且对时延标准较为敏感的数据。在媒体和娱乐行业的海量需求推动下，StorNext 不断发展，始终保持领先一步。如今，StorNext 的高性能文件存储已成为行业内用于存储、传输和共享视频和类视频数据内容的标准选择。出于其出众的性能和灵活性，它也适用于许多其他类型的数据和工作负载。

StorNext 作为 CH 出品的文件系统。不仅能在 CH 硬件设备上实现简易化管理和快速部署，也可以与其他品牌的计算和存储硬件实现完美适配。StorNext 广泛的可配置性和协调性，使其无论部署在任何平台上，都可以实现潜力的最大化。

在对 StorNext 存储进行了一些基本的定位之后,本文将解释 StorNext 存储的体系结构,描述 StorNext 存储的诸多特性与功能，重点介绍 StorNext 存储那些独具特色的技术细节和专利技术。

StorNext 存储体系架构-概述

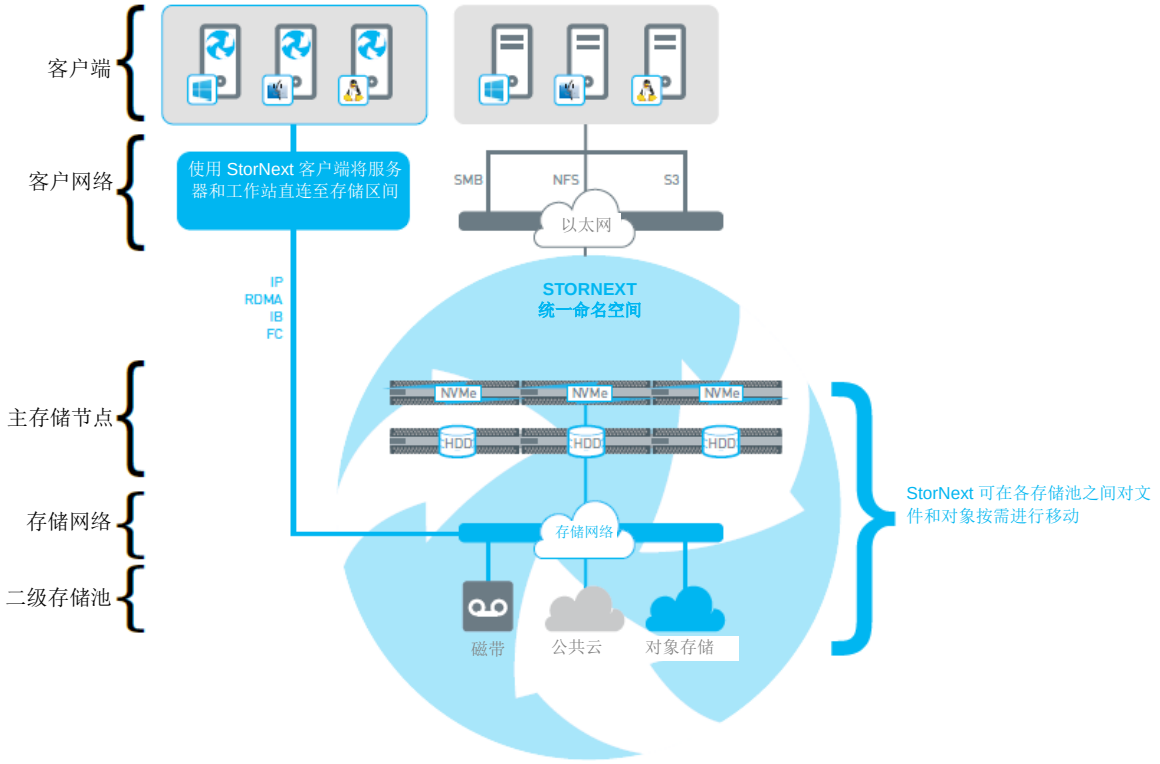


图 1-StorNext 存储架构

StorNext 存储系统包含几种不同类型的组件，如图 1 所示。核心是运行 StorNext 数据服务的，包含计算、存储或两者兼具的聚合节点。在最小的系统中，StorNext 也可以仅存在于单个节点中，但是节点集群是最为常见的，因为其可以提供更高的冗余和规模。

在节点上运行的 StorNext 数据服务，包括数据副本的管理和存储的最终介质（包括闪存、磁盘、磁带、对象存储和云空间）的策略引擎功能。无论数据是被保存在此体系结构中的任意位置，其都是可见的，并且命名空间内的其他客户端是可对其进行访问的。策略引擎支持可用于存储性能、成本和保护级别的自动优化。

StorNext 客户端为可访问一个或多个 StorNext 文件系统的服务器和工作站。客户端间有若干种连接方式，每种方式都有其特定的优点。从使用的便捷性和普及性方面来说，SMB 和 NFS 为理想之选。尽管可以使用 NAS 协议来实现内容的流式传输，但这并非其设计的预期用途。为了获得最高性能，系统可以通过 StorNext SAN 客户端软件来进行连接。运行 StorNext 客户端软件的服务器和工作站可与 StorNext 文件系统建立高度优化的直连。甚至还可以提供 S3 协议供前端访问。因此，可将部分 StorNext 系统容量留出，供任何使用 S3 协议中 PUT 和 GET 的应用程序用作对象存储。

这些组件由两类网络进行连接：*客户端网络*将客户端连接到文件系统和其中的存储空间。*存储网络*将节点连接在一起并连接到外部存储的介质。运行 StorNext 客户端软件的设备也可连接至存储网络，这是其特殊关键功能之一。

STORNEXT 文件系统

StorNext 存储系统的核心是 StorNext 文件系统，又名 SNFS。StorNext 存储的强大性能、可扩展性和灵活性，在很大程度上归功于 SNFS 的功能。文件系统的编写非常复杂，风险也很高。如构建良好，文件系统可以摆脱所有存储方面的困扰，在保持数据的一致性和完整性的同时实现最高的性能。如构建不当，文件系统可能会影响性能，甚至导致数据丢失。几十年来，SNFS 一直在不断地发展、细化并改进，其安全性和超高性能已经在世界上许多条件苛刻的数据环境中得到了证明。

StorNext 系统包含一个或多个 StorNext 文件系统，这取决于客户的需求和目标。图 2 说明了组成 StorNext 文件系统的主要组件，及其与底层存储的关系。

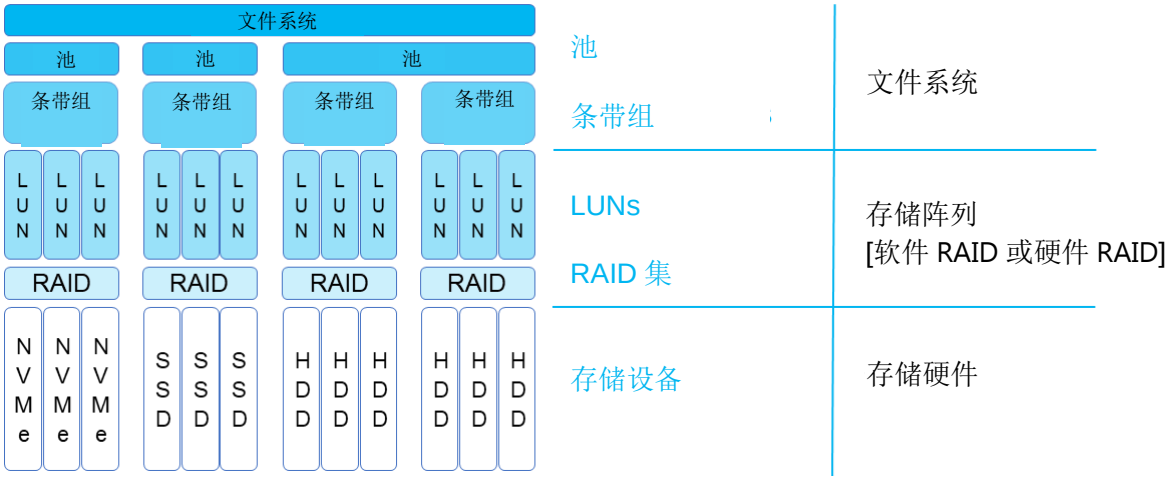


图 2-StorNext 文件系统架构

存储设备

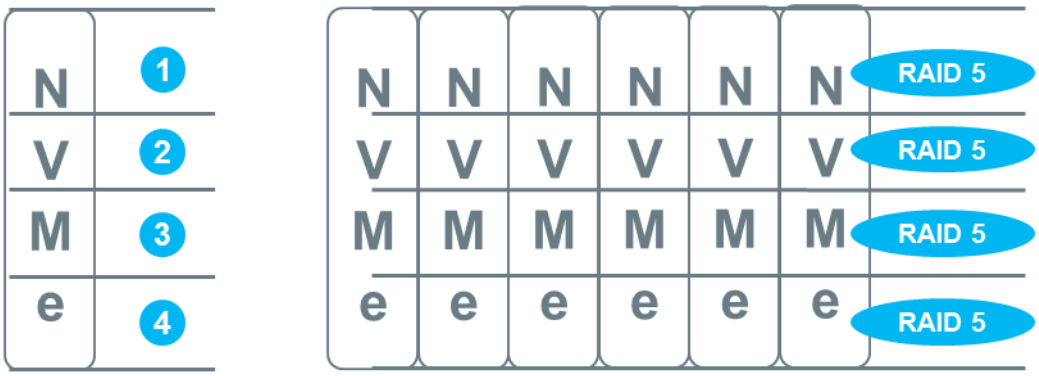
SNFS 存储与设备无关，也就是说 SNFS 支持多种类型、多种品牌、多种接口的存储设备。存储系统是否包含 HDD、SSD、NVMe 或内存形式存储并不重要。只要连接端的设备为块存储，SNFS 就可以使用它。还可以在同一文件系统内组合多种类型的存储，并保持其各自的性能特性。

RAID 集和 LUN

为了提高性能和冗余，磁盘阵列通常会将原始存储设备聚合到 RAID 集中。这可以由专用硬件存储控制器完成，或完全通过存储系统中的软件完成。

一旦创建了 RAID 集，就定义了逻辑单元编号（LUN）。根据配置的不同，RAID 集和 LUN 之间可能存在 1:1 的对应关系，或者单个 RAID 集可以被分割成多个较小的 LUN。例如，可将两个驱动器的 RAID1 镜像定义为单个 LUN，并且可将 12 个驱动器的大型 RAID6 集划分为四个较小的 LUN。

企业级 NVMe 驱动器提供了一个额外的配置选项，使用命名空间，类似于 HDD 上的分区。RAID 集和 LUN 可以跨几个 NVMe 驱动器的完成创建，如图 3 所示。由于 NVMe 具有大规模并行可访问性，因此这种方法比较适用于 NVMe，而对于 HDD 分区，它会降低驱动器性能并缩短设备使用寿命。



每个驱动器被分成四个“命名空间”-类似于分区。

在此示例中，5+1 个 RAID5 集在一组六个驱动器上被条带化；每个命名空间一个 RAID 集。

图 3-使用 NVMe 命名空间的 RAID

条带组

简介

存储设备、RAID 和 LUN 在块存储系统中屡见不鲜。进入到文件系统结构中，情况会变得更为细化。传统的文件系统需构建在一组 LUN 或磁盘分区等其他逻辑设备之上。StorNext 在 LUN 和具有重要扩展功能的文件系统之间引入了两个抽象的层：条带组和池。

条带组可被简单定义为一组选定的无差别 LUN。数据在这些 LUN 之间被条带化，形成可配置参数，包括：

- **段大小（Segment Size）：**在将数据写入该 LUN 中的下一个驱动器之前，写入 RAID LUN 中的一个驱动器的数据量。这是为匹配 RAID 阵列软件或硬件的配置。
- **数据条带宽度（Data Stripe Breadth）：**在切换到条带组内的下一个 LUN 之前，SNFS 向 LUN 写入的数据量。
- **Inode 条带宽度（Inode Stripe Width）：**如果不为零，则会导致大文件的分配，是以指定大小块的形式跨条带组条带化。

当使用自动条带组配置时，需在后台设置许多附加参数。手动配置模式可让用户自己调节所有参数。完整的技术细节可在 StorNext 文档中心中查阅。

条带组的类型和用例

SNFS 可使用条带组将具有不同特性的数据分离到不同的 LUN 上。SNFS 的条带组有三类：

- **元数据**条带组保存的是文件系统元数据，包括文件系统中每个文件的文件名和属性。元数据非常小，可以随机访问。
- **日志**条带组包含文件系统日志，即对文件系统元数据的更改的顺序记录。日志数据是一系列小的顺序写入和读取。
- **用户数据**条带组包含文件的内容。用户数据的访问模式取决于客户的具体场景，但视频文件（一个常见的例子）是按顺序访问的大型文件。

在性能要求相对较低的较小系统中，所有三种类型信息均可驻留在一个条带组上。典型的高性能配置可以将元数据和日志数据与单独的用户数据条带组组合在一起。在非常庞大和非常繁忙的系统中，通过将元数据、日志和用户数据分离到单不同的条带组上，可以获得额外的性能。

对于可以存放多个用户数据的条带组，每个用户数据条带组会针对不同类型的数据或访问要求进行优化。例如，视频播放需要具有一致的、低延迟的高性能流，从而可避免帧中内容的丢失。而渲染作业涉及更具随机性的访问、更小的 I/O。通过 StorNext，您可以使用不同的一个或多个条带组来存储并优化不同的数据，并且仍然将它们作为同一文件系统的一部分。

由于条带组中的 LUN 无需都被保存在同一存储系统上，因此条带组还提供了一种可扩展文件系统性能和容量的方法。为实现扩容目的，条带组可以跨越两个大容量存储，每个存储都包含许多个扩展盘柜。为了实现性能扩展，条带组可跨越多台容量较低的存储，以利用聚合存储的控制器带宽。这两种技术都在图 4 中进行了可视化描述。

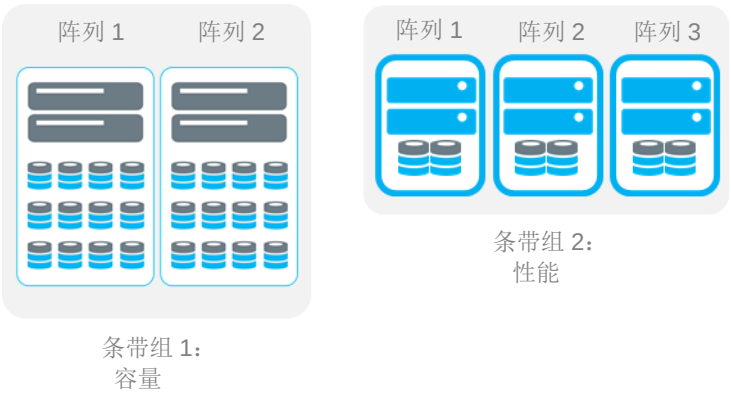


图 4 -使用条带组扩展容量和性能

将新的条带组添加到现有文件系统后，配合使用精简配置的存储设备，可以增加条带组内 LUN 的大小。*卸载*功能甚至允许文件在后台的条带组之间完成迁移。还可以添加新的存储设备，并在系统在线时停用旧的存储设备。

最后，条带组还支持更细粒度的数据布局。在文件系统内，特定目录或甚至特定文件类型（例如，后缀为.jpeg）可以存储到特定条带组。这些联系被定义为*亲和性*。这种亲和性能够利用不同条带组的独特性能特征，同时在单个文件系统中将所有文件按逻辑分布连接在一起。

存储池

简介和用例

SNFS 存储池是 StorNext 文件系统中的第二层抽象的概念。虽然条带组是强制的，但存储池是可选的。再次参考图 2 可发现，存储池是由一个或多个条带组组成的，就像条带组由一个或多个 LUN 组成一样。每个存储池都存在一个关联的命名空间，如“metadata”或“journal”。

存储池允许在不同类别的主存储器之间完成数据的“透明”转移，例如 NVMe 闪存池和 HDD 池。这种转移需要由管理员手动操作，也可由预定义的策略自动触发。“透明”指从客户端的角度来看，文件总是保留在 StorNext 命名空间中的原始位置，而无需关心它们实际驻留在哪个储存池中。

一些情况下，文件在其生命周期的不同时期需要不同的存储性能，而不同的性能也代表着不同的存储需求，存储池的价值在于可大幅提升数据在实际应用中的效率。例如，在非线性视频编辑任务中，特别是在较高分辨率和帧率下，需要非常高的带宽和极低的延时。这是一个完美的 NVMe 存储应用场景。但就单位容量成本计算（$/TB），NVMe 仍然比 HDD 存储设备贵得多。存储池可随时随需地利用少量的快速存储（如 NVMe）存储，为部分不需要极佳性能的工作流或数据生命周期阶段提供更廉价的大容量存储（比如 HDD）。

存储池的另一个用途在于可以将不同的项目内容隔离起来。这样，对某一个项目文件的高频使用并不会影响其他项目的用户的性能体验。存储池也适用于高性能摄取场景，在该场景中，数据如果生成于高速存储池中，随后可以自动转移至到更便宜、速度稍慢的次级存储中，从而为新内容腾出空间。

作业和策略

在存储池的概念中，作业是系统对特定文件执行操作的一组指令，可立即运行，也可在将来特定的时间运行。*策略*是定期运行特定作业的一套规则。另外，策略的触发运行可设置先决条件，比如仅当池的剩余容量高于指定水平 （例如，存储量已超过总容量的 80%）；策略也可以用于限制在每次运行所执行的工作量（例如，每次运行，可处理最高 1TB 的工作量）。

要操作的文件集可以显式指定，也可以基于一组特定的标准指定，标准包括文件大小、创建历史文件或目录、驻留池的任何组合，甚至是名称是否匹配自定义正则表达式（regex）。一个示例策略如下，当名为“fast”的存储池中超过一周的文件填充级别超过 80%时，在文件系统中搜索该类文件并将其移至名为“slow”的池中。

除系统管理员外，还可向其他用户和组授予*用户*或*管理员*访问权限，从而完成作业和策略的提交和控制。用户可控制被授权的作业以及查看相关的存储池与策略。管理员可执行包括配置在内的任何操作。

下列为几种不同的作业类型：

作业类型	操作
移动(Move)	文件内容从一个池移动到另一个池
预估(Estimate)	可选择在移动前运行，以计算目标池中所需空间
升级(Promote)	在目标池中创建内容的新副本并保留旧副本
降级(Demote)	如果升级的内容没有更改，请删除升级的副本，恢复原文本。对于已更改的文件，将更改后的副本移动到最开始的存储池中
移除(Remove)	移除升级的内容，保留原文本不变
盘点(Inventory)	对每个池中的文件和目录进行计数并提供报告
校验(Checksum)	使用指定算法对每个文件进行校验和，然后将校验和结果存储在文件的扩展属性中。使用者可从外部访问存储的校验码，系统也支持自定义集成
验证(Validate)	使用指定算法读取并执行每个文件的校验和，与存储的校验码进行比较，并及时报告任何不匹配的情况

表 1-存储池作业类型汇总

存储池可灵活高效地控制数据在 StorNext 系统的主存储环境内的保存位置，并且还可与稍后描述的 FlexTier 二级存储分层实现兼容。

扩展

数据传输是通过系统服务执行的，并且可通过在集群中多个节点上运行服务，来轻松扩大池的运行容量。在图 5 的示例中，mover 服务在集群中的所有三个节点上运行。两个实例配置为在文件系统“FS1”上操作，另一个实例配置为在文件系统“FS2”上操作。

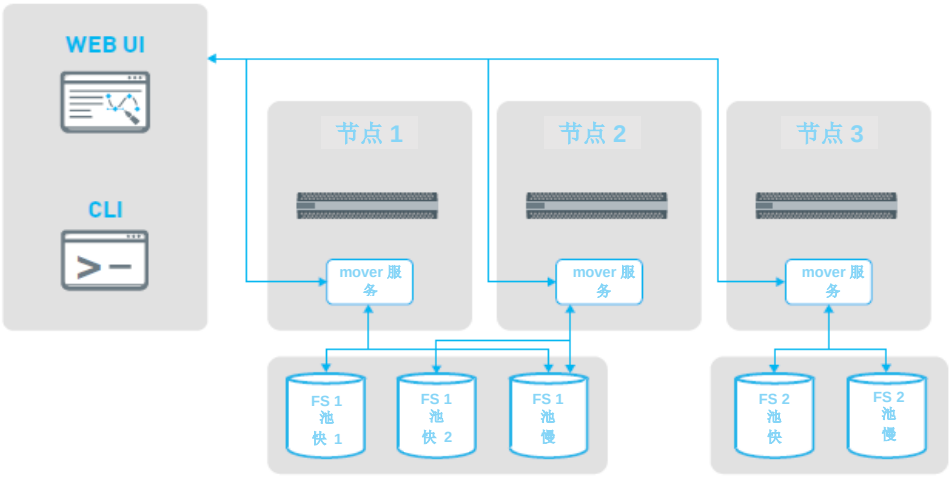


图 5-扩展池传输服务

文件系统

我们已详细介绍了 StorNext 文件系统的基本结构，接下来会主要介绍一下文件系统本身的一些主要功能与关键特征特征。StorNext 文件存储系统集群可根据需要，并发运行多个文件系统，所有文件系统均具备唯一配置。

元数据

在文件系统中，元数据（字面意思是“关于数据的数据”）起着关键作用。它由文件名和位置信息组成，同时还包括时间戳、权限、只读和脱机指示符等标志以及许多其他属性。像 StorNext 这样的共享文件系统，还可以解决元数据一致性方面的挑战，确保所有用户看到的文件系统是一摸一样的。

本文中提到的“元数据”，特指上一段所述的文件系统元数据。这与内容元数据不同，比如创建者、音轨号、视频分辨率、比特率等。StorNext 可对文件系统元数据实施管理，而像媒体资产管理系统或其他内容管理器（例如 iTunes）等外部系统，不会受到 StorNext 管理内容元数据的影响。

文件系统元数据是文件系统性能的关键。如果元数据操作较慢，则会限制创建或更改文件的速度。在 StorNext 文件系统中，元数据与文件数据是分开处理的。前者通常被保留在经过调优的、可对元数据进行优化使用的单独存储空间内，但小型系统除外。同理，元数据通信也可在与数据分离的网络上进行，并且通常发生在高性能系统当中，有助于减少元数据通信的延迟。

StorNext 在元数据通信中的高效技术已获得多项专利保护，包括 8190736、8554909、9456041 和 9342568。元数据通信的相关内容，将在本文档的 StorNext 客户端部分作进一步分析。

元数据通常被缓存在内存中，避免从 HDD 或 SSD 存储中检索元数据时出现延迟。在储存有数百万个文件的文件系统中，元数据的量同样也会很大，因此很难将其缓存在适当大小的 RAM 中。StorNext 中的关键元数据性能的主要一项创新之处在于，其具有多级元数据缓存，包括专利 （9176887) 压缩二级缓存技术。高效的压缩可以使数亿级别，甚至更多文件的元数据均可以缓存在集群节点上的 RAM 中。除了最大的文件计数环境之外，在所有环境中，元数据读取操作均可从 RAM 中完成且不必访问速度较慢的存储空间。

专利 8977590 中描述的智能缓冲技术，加快了 StorNext 文件系统中，元数据到内存的写入速度。通常来说，公共缓冲对写入设备是透明的，如果在将缓冲数据写入存储前发生故障情况，则缓冲数据可能会被丢失。

通过 StorNext 的智能缓冲，写入客户端时会提示使用者，哪些元数据正处于缓冲中，当缓冲器发生故障或出现丢失情况下，客户端将重新发送元数据，从而避免任何数据丢失的可能。

使用包括专利 9069790 和 9483356 所涵盖的技术，可以对文件系统日志进行类似的优化，从而获得最高的性能和效率。

StorNext 元数据处理中最有力的一项创新便是 StorNext 元数据存档（MD 存档-metadata archive）。MD 存档是用于存储元数据历史的自定义空间数据库（专利 10133761）。得益于数据库的性质和其中元数据的编码方式，可实现快速、高效的搜索。如果没有此类型的数据库，一些操作将需要通过扫描完成，例如在文件系统的一个区域中查找更改。对于大型文件系统，扫描会消耗大量的系统资源和时间。StorNext MD 存档通过采用简单查询替换扫描，使复制和碎片整理等操作，能够以较低的资源成本快速运行。由于它还包含文件系统的更改记录，所以元数据存档可用于审核或取证，从而确定“A 对 B 实施了哪些操作，以及操作时长等信息”。当然，使用者也不能完全弃用扫描操作，因为扫描是验证文件系统和 MD 存档是否同步的一个重要方法，但扫描的使用频率，比起上述情形，会大大的降低。需扫描时，可同时使用特殊的并行技术（专利 14922432）实现快速高效的扫描操作。

分配

确定数据的存储位置，即解决数据的分配问题，是任何文件系统需要实现的又一项重要工作。分配的合理与否会直接影响系统的各项性能，包括数据是初始写入性能、读取性能以及文件系统在一段时间内的性能状况波动，还会影响碎片化文件的产生概率。而 StorNext 在该领域，同样具有重要的知识产权。

SNFS 可以“积极地”去进行空间分配。假设它对某个文件执行了写入操作，则以后很可能还会对该文件执行另一次写入操作。为将文件数据保存在一起，并防止数据和剩余空间的碎片化，分配的空间一定要比请求空间大。例如，如果写入 1MB 文件，则可分配 2MB。最初分配的额外空间量取决于文件大小，并且会以递增的方式扩大至指定上限。下列图 6 示例中，第一个分配为 2MB，第二个为 6MB（2MB 加 4MB 增量），第三个是 10MB（2MB+2x 4MB 增量）。

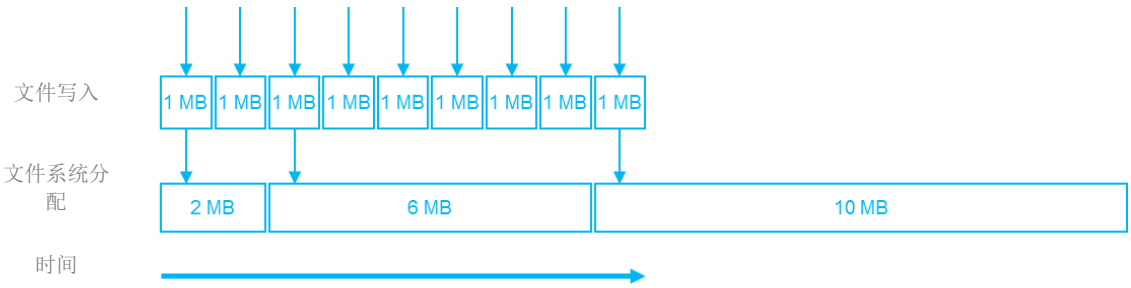


图 6 -优化的空间分配示例

可根据需要，轻松优化分配算法，从而针对特定写入模式进行优化。系统提供了查看分配统计信息的工具，可辅助进行调优。关于分配优化的更多细节，请查阅此处的在线文档。

批量预分配（Allocation Session Reservation ，ASR）是一项专利[8271760]分配技术，可防止文件系统的碎片化，提高在特定目录中写入和读取文件序列用例的性能。这种模式常见于富媒体流应用，及其他的相似应用。启用 ASR[默认]后，目录中的文件序列将按写入顺序存储在磁盘中。ASR 会将这些文件保存在一起，即便其他应用程序同时存在于不同的目录中，或是从不同的客户端写入的。

如果未启用 ASR，在对多个客户端或应用程序同时写入文件序列时，可能导致存储出现“棋盘化交错”，即不相关的数据被交叉存储。在读取文件时，棋盘化交错会对性能产生严重的负面影响。因为这些应用程序会按顺序写入和读取多个文件，从而使得单个文件连续化的简单“碎片整理”操作不起作用。ASR 可防止相关文件在写入时出现整个序列碎片化的情况发生。除可减少文件系统上的数据碎片外，由于一起写入的文件集合通常会一起移除，因此 ASR 还可减少自由空间中的碎片。

图 7-分配会话预留示例

由于每个客户操作数据流的方式是唯一的，StorNext 包含了一系列的特性和工具，既可防止碎片化的出现，又可管理任何文件系统在长期使用时不可避免发生的碎片化情形。这些工具的摘要位于此处。

当应用程序执行的 I/O 与配置的 RAID 条带大小不匹配时，系统性能同样会受到影响。在某些情况下，这是不可避免的，比如处理像 DPX 这样“每帧都作为一个文件”的视频格式时。RAID 软件会计算全条带上的奇偶校验。当写入“很短”时，RAID 软件必须首先从存储中读取条带中的其余现有数据，然后再计算新的奇偶校验。所产生的延迟和额外的磁盘 I/O 会导致明显的延迟情况。StorNext 通过使用一种有助于填充较短写入任务的独特策略（专利 8650357），避免了在该情况下可能另行发生的额外读取。

安全模型

在现代文件系统中有两种主要的安全模型：UNIX 权限位和访问控制列表（Access Control Lists ，ACL）。StorNext 同时支持这两种模型，但何时使用它们，取决于客户端的 OS 和 SNFS 配置设置。根据每个文件系统可对安全模型进行灵活选择。具体使用哪个模型取决于其的安全性要求。UNIX 权限位较为简单，但在灵活性方面略逊于 ACL。这两种模型都允许在所有平台上，统一实施单类型权限，从而在异构环境中提供一致的体验。

通过 ACL 安全模型，可在基于 ACL 的 Windows 系统上进行权限管理。在 Linux 和 Mac 平台上，安全检查可基于 ACL 和 UNIX 权限位的组合。在 UNIX 权限位模型中，可根据 UNIX 权限位在包括 Windows 在内的所有 OS 平台上实施权限管理。对于 Windows，UNIX 权限在 Windows 资源管理器中会被显示为合成 ACL，其中一个访问控制项（ACE）可用于操作用户，另一个用于组，还有一个用于授权给所有人。

有关 StorNext 安全模型的详细信息，包括标识映射、跨平台权限执行和其他主题，请参阅 changhongit.com 上 StorNext 文档中心的 StorNext 安全部分。

StorNext 客户端与连接性能

简介

使用由 StorNext 集群提供共享存储服务的机器和进程，通常被称为客户端机器，简称客户端。SNFS 作为一个异构文件系统，因此客户端可自由运行 Windows、macOS 或 Linux 系统。客户端计算机可通过任意形式的客户端软件，连接至 StorNext 集群，这些客户端软件可以是内置在 OS 中的，也可单独安装。每种客户端连接方法的显著优点及特征可见下方表 2。StorNext 环境，包括了需要使用不同连接方法的各种客户端组合。

客户端连接通过	性能	储存路径	客户端软件	数据网络
NAS （NFS/SMB）	好	NAS 网关	无需安装，操作系统默认提供	标准以太网
StorNext 客户端软件 — 网关模式	高	StorNext 网关	单独安装*	标准以太网
StorNext 客户端软件 — 直接模式	最高	直接	单独安装*	光纤通道、IB、iSCSI、RDMA 以太网
S3 客户端软件	好	S3 网关	单独安装	标准以太网

**macOS 包括一个 StorNext 客户端软件版本 XSAN。Windows&Linux 需安装 StorNext 客户端软件。*

表 2-StorNext 客户端概要

下方的 StorNext 架构图为各种客户端和客户端连接等软硬件组件的详细视图。

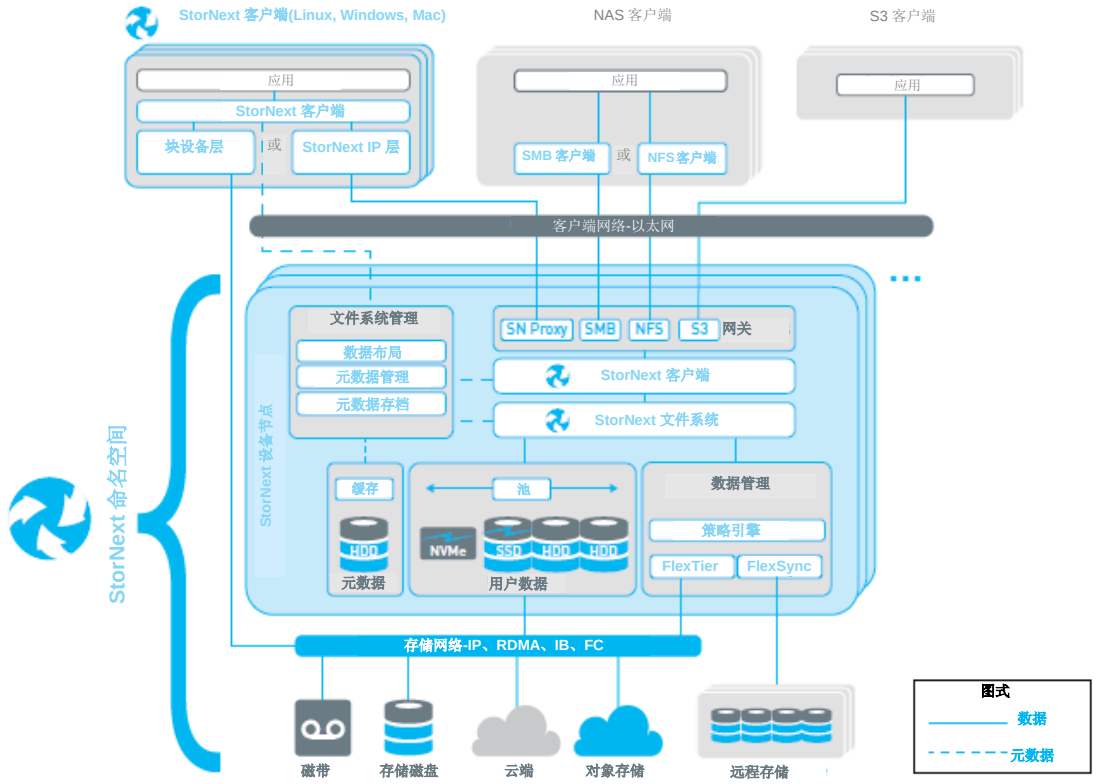


图 8-StorNext 详细架构

NAS

NAS 是客户端的首选连接选项，使用标准 NFS 或 SMB 协议。由于现代操作系统中普遍包含连接至 NAS 共享的功能，因此 NAS 连接过程极为简单且无需安装代码。它可通过已设置的 TCP/IP 网络进行工作，且速度相当快，适用于各类用户及用例。然而，对于涉及大型文件，或对时延敏感的数据流的高性能工作负载来说，这显然还不够好。

NAS 在这些应用中表现不佳，原因之一在于，它对网络带宽的使用效率不高。传输的单位数据存在大量协议负荷。另一个原因是 NAS 客户端和服务器软件进行了一些折中设计，因此在大多数情况下，可为多数用户和用例提供良好性能。但经优化设计后，其也可为大量用户和相对较小的请求提供支持。

在 StorNext 集群中，可将一个或多个节点配置为 NAS 网关。如果配置了多个节点，网关可以通过内部 DNS 服务器和虚拟 IP 地址（VIP），以冗余和负载平衡的扩展方式完成操作。如果某个节点发生故障，现有连接将被移至其他活动节点。如果某个节点出现联机返回或整体环境中被添加了某个新节点，连接将自动进行重新负载均衡。

图 9 显示了一套八节点的 StorNext 集群，其中四个节点配置为 NAS 网关，VIP 用于负载平衡和故障转移。客户端将始终连接集群 VIP 上的集群（此处为 10.1.1.1），主节点负责负载分配。如果主节点发生故障，则主任务将被移至其他活动节点，并且横跨剩余节点重新进行负载均衡连接。

有关 StorNext 扩展 NAS 集群操作的更多详细信息，请参见 StorNext 文档中心。

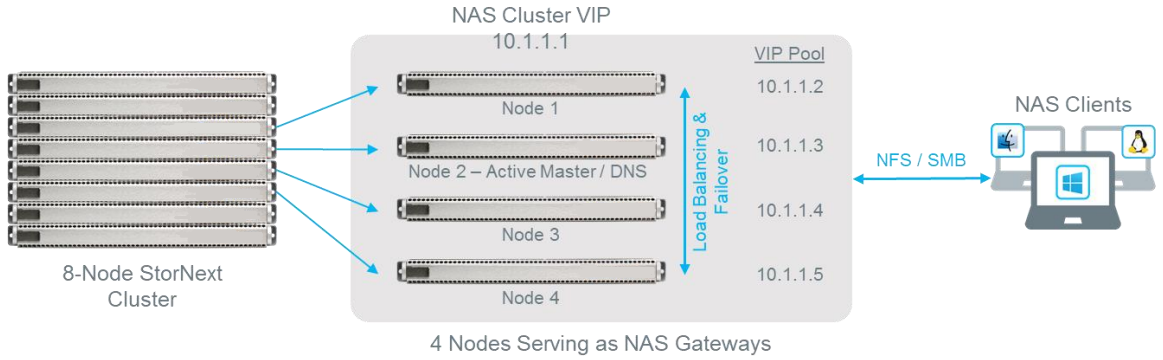


图 9-StorNext NAS 集群

StorNext 客户端软件

连接到 StorNext 集群并实现最高性能的方法，是通过 StorNext 客户端软件。苹果设备可以利用 MacOS 内置的 XSan 客户端；Windows 和 Linux 设备需完成简单的客户端安装。

当使用 StorNext 客户端软件时，已挂载 StorNext 的文件系统，将在客户端机器上显示为本地文件系统，而不是远程 NAS 挂载，如图 10 所示。StorNext 文件系统是为远程共享而设置的，这一点对于应用程序来说，是透明的。

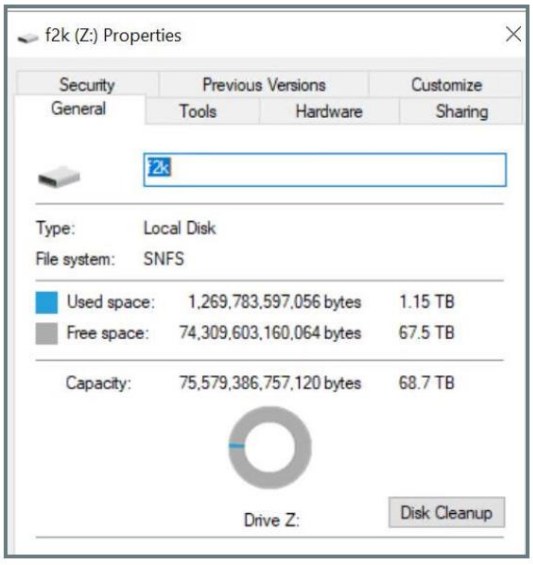


图 10 - Windows 客户端上的 SNFS 属性页

除透明性外，通过 StorNext 客户端挂载的 StorNext 文件系统性能，要比通过 NAS 挂载的性能高得多，特别是对于大型的文件和数据流。StorNext 客户端可控制连接的两端，因此可通过使用更有效的传输方案，让可传输的数据规模变得更大，而对应的带宽开销变得更低。StorNext 客户端和服务器端软件专为实现流式应用中的最高性能而设计，而这是传统 NAS 所无法实现的。

连接模式详情

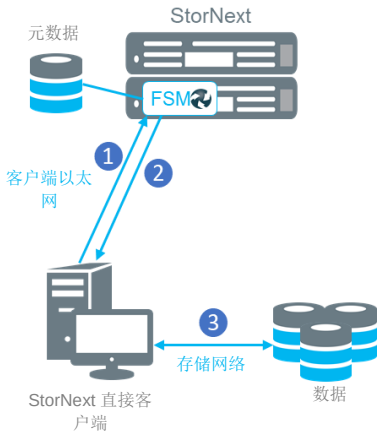
StorNext 客户端软件可以直连模式或代理模式运行，区别在于客户端是如何连接到共享块存储区间的。以直连模式运行的客户端称为直连模式客户端，简称直连客户端。以代理模式运行的客户端称为代理模式客户端，简称代理客户端。图 8 直观说明了以下文本所述技术细节。

直连模式

直连模式是连接到 SNFS 的最初方法，也是客户端和块存储间最短、最快的路径。直连客户端通过后端的存储网络，直连至共享块存储，后端存储网络可通过光纤通道、RDMA、iSCSI 或 InfiniBand 实现。读写活动只涉及客户端和存储器，没有类似“NAS 存储引擎”或其他笨重的软件堆栈。除将客户端连接至存储器的存储网络外，这与服务器写入本地内部存储的方式十分类似。

然而，与服务器中的本地存储不同，StorNext 文件系统后的存储为共享存储模式。我们需通过特定的方法来确保共享存储的一致性，因此，所有客户端始终会装有相同文件系统的一致性视图。对于 StorNext 来说，这个功能由文件系统管理器（FSM）执行。除经由存储网络连接到块存储外，直连模式的客户端还可维持 TCP/IP 与 FSM 的连接，从而实现元数据通信。FSM 可谓是确保文件系统一致性的“交通警察”。

图 11 为直连模式 StorNext 客户端、FSM 和存储间实现读取和写入操作的数据和元数据流量的简化视图。需注意的关键方面是 FSM 并不在数据路径中。一旦 FSM 向客户端传达存储上相关块位置的信息，客户端会直接执行块存储的数据读取和写入工作。



序号	读取	写入
1	ABC 文件的数据位置	写入位置
2	FSM: 块 XX 至 YY	FSM: 块 AA 至 BB
3	直接从存储读取数据	将数据直接写入存储

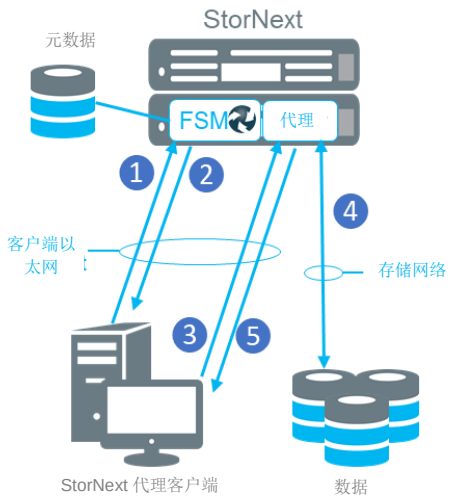
图 11-StorNext 直接客户端通信流程

直连模式的优点是速度快，缺点是成本高。连接至高速存储网络的成本要高于连接至前端客户端网络的成本。然而 NAS 对于许多应用而言，传输速度不太理想。幸运的是，我们还有第三种选择方案。

代理模式

代理模式（以前称为 DLC）传输速度比 NAS 更快，且网络连接成本更低，而且具有独特的负载平衡和可扩展特性，因此理想适用于以下应用要求：需要实现 NAS 无法实现的高性能和低时延，但也无需达到直连模式提供的最高级别性能。

与直连客户端不同，代理客户端不会直连至存储网络，相反，它通过在 StorNext 集群上运行的网关对数据进行代理处理。代理网关可像直接客户端一样，直接访问块存储。当代理客户端需执行 I/O 时，首先它会通过指定的元数据网络与 FSM 建立联系，获得存储器上的位置信息。然后代理客户端会将数据请求传输回代理网关，代理网关代表代理客户端对存储器进行访问，并传输回所需的数据和状态。该过程如图 12 所示。



序号	读取	写入
1	ABC 文件的数据位置	写入位置
2	FSM: 块 XX 至 YY	FSM: 块 AA 至 BB
3	将块 XX 读取至 XX+n	将块 AA 写入 AA+n
4	代理提取数据块	代理写入数据块
5	代理将块返回给客户端	代理向客户端确认

图 12-StorNext 代理客户端通信流程

在客户端和存储之间使用网关听起来就像 NAS 的运行方式一样，那么代理客户端该如何实现更高性能？这一问题的答案涉及几个方面。

- 与典型的 NAS 堆栈相比，代理客户端软件堆栈非常精简，并且 CPU 消耗也少得多。
- 代理客户端使用的唯一协议比 SMB 或 NFS 简单得多。
- SMB 的大多数的实现，都是每个客户端单线程处理元数据请求的结果，限制了性能
- 甚至 SMB 多通道也不能跨节点拆分 I/O，因此性能受限于单个 NAS 节点的性能。
- 代理客户端负载均衡功能，是其使用的唯一通信协议中固有的。NAS 负载平衡基于 DNS 实现，因此会导致更高的延迟。
- 通过 StorNext 代理客户端，即使针对单个客户端和单一数据流，也可实现更高等级的并行性，具体如下。

体系结构中的并行性，是代理模式客户端实现可扩展性、负载平衡和弹性等特性的关键。通过将客户端上的大型 I/O 分割成较小的并发 I/O 请求，这些请求可跨越多个代理网关节点发出，然后通过并行，使用多个网卡提高吞吐量，单个的请求也是实行同样的操作。为了更好地优化超高速（40/50/100GbE）链路的性能，每个网卡上都配置了多个并行连接，从而可以更好地利用多核 CPU。

代理网关可以基于队列的深度来平衡负载。如果到特定网关节点的 I/O 队列开始增长，但完成得不够快，代理客户端就会将随后的 I/O 转移到不同的网关节点上。该机制实现了基于网关节点时延和吞吐量的实时负载平衡，并避免单节点出现性能堵塞的状况。处理请求较快的节点将接收更多客户端请求。当所有队列都为空时，各个代理模式客户端会随机挑选单个网关节点，来完成服务请求。

当添加额外的网关节点时，代理客户端会被自动告知此次添加行为，同时可以将这些网关节点用于 I/O 请求。如果网关节点不可用，代理客户端将继续使用剩余网关。并将发生故障的网关上进行的 I/O 自动重新发布。

在一些更为复杂的环境下，某些客户、或某组客户为确保优异的性能的始终如一，可能需要一定程度的手动控制。为此，可将客户端配置为使用网关节点，或网卡的特定子集，从而可在不牺牲负载平衡和弹性的情况下，完成资源分配。

S3

第三种连接到 StorNext 文件系统的方法是通过 S3，可使用任何支持 S3 的应用程序实现。此方法可以使 StorNext 能够写入一个具有某些独特特性的，简单的对象存储系统。我们会在后文中讲到，凭借 StorNext 策略引擎将数据写出到对象存储的功能，请注意这两种功能是不能混为一谈的。

对象数据驻留在文件系统的专用子目录中，例如/ObjectStore。该根目录下的结构是不可读的，因此对象存储目录不会与 NAS 客户端、或运行 StorNext 客户端软件的客户端共享。对象存储数据将仅通过 S3 接口进出。

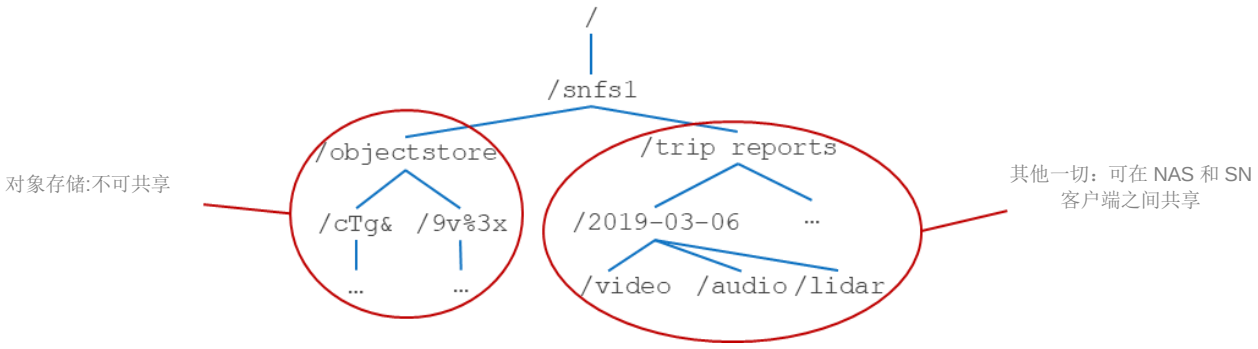


图 13-对象存储目录共享与其他对比

通过指定的集群节点访问对象存储，这些节点可充当 S3 网关，从而提供与外部应用的高可用性连接。可根据需要配置多个对象存储，每个对象存储都可配置对应的 S3 网关集群。

StorNext 对象存储由对象和容器的层次结构组成，如图 14 所示。单个对象存储在存储桶中，其是一个平面的、非分层的命名空间。一个项目中可包含多个存储桶，一个对象存储中也可能包含多个项目。

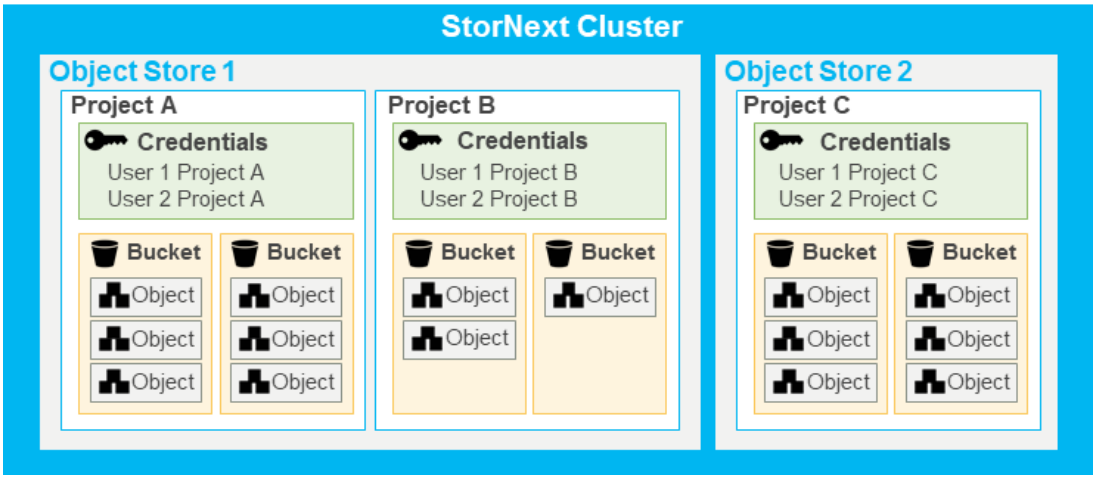


图 14-对象存储结构

对象存储的安全性通过用户和项目的凭据认证管理而实现。用户的管理可通过以下方式进行：与 Microsoft AD 连接；使用 StorNext 集群上的 LDAP 服务器维护的本地用户账户。每个用户对其访问的每个项目都持有唯一凭据。如果这看起来似乎不太寻常，那是因为它确实如此，至少与标准文件系统的运行方式相比，这一体系是非常独到的。请记住，出于速度和可扩展性的考虑，对象存储的设计，采用了非常简单的规模语义，并且可与应用程序直接完成交互，无需手动介入的操作。

将 S3 访问与 StorNext FlexTier 数据管理策略相结合，可将对象从磁盘“分流”到磁带以实现长期归档存储，因为归档存储比公有云的冷存储更具成本优势。

数据服务

除高速文件系统外，StorNext 还可以完成大量的数据集成服务。有些可用于日常存储资源的管理，有些则被用于诸如数据保护，数据生命周期管理，或与其他系统之间的数据导入/导出等功能。

虽然 StorNext 数据服务可由管理员手动控制，其也可由外部应用，以编程方式进行控制，但该服务通常是通过 StorNext 的集成策略和调度功能完成驱动的。例如，存储策略可指定，目录/ data 中的文件在被更改五分钟后，将其副本自动复制到本地的对象存储中，并将另一个副本复制到公有云当中，并在进行进一步的更改后，保留 10 个版本。12 个月后，本地副本自动过期。

刚才描述的简单策略，实现了全面的现场和非现场数据保护，并在文件老化时自动将其移动至云端，从而达到控制存储成本的目的。

表 3 包含 StorNext 数据服务的概要，下文将对其进行详细描述。

数据服务	说明	用途
数据迁移	在条带组之间移动文件	资源管理
配额	对文件系统容量的共享进行控制	资源管理
QoS/带宽管理	对文件系统性能的共享进行控制	资源管理
FlexTier	将 StorNext 文件系统以透明方式扩展到二级辅助存储	资源管理 数据保护 数据分发
FlexSync	目录复制/同步	数据保护 数据分发
导入/导出	从外部系统导入文件或将文件导出到外部系统（磁带、对象存储、其他存档 SW）	数据交换 数据迁移
网络服务 API	支持与外部应用程序（如媒体资产管理器）的编程集成	应用集成

表 3-storNext 数据服务概述

数据迁移

虽然 StorNext 存储系统可根据不断变化的需求进行扩展、和更改，但其架构下的硬件组件始终会面临过时、淘汰的风险，对于存储设备而言，尤其如此。随着更高性能和更具成本效益的技术的出现，使用者通常需要每两到三年更换一次存储阵列。

通过 SNFS，可使用文件系统扩展，添加配置为一个或多个新条带组的附加存储。添加了新的条带组后，有两种数据迁移的选项：既可将来自源条带组的文件重定向到特定目标条带组，也可通过操作，将来自一个或一个以上源条带组的文件，分散到所有剩余条带组之间。

在这两种情况下，迁移过程首先会将源条带组的分配状态（alloc）设置为 false，从而防止创建新的文件扩展区或扩展现有扩展区。如果将源条带组简单地标记为只读，则会造成应用程序的写入失败。使用 alloc 标志可避免这种情况。

一旦在源条带组上完成了 alloc=false 的设置，就可以在文件系统处于活动状态时，在后台迁移文件内容。如果打开源文件，则会通知客户端暂停 I/O，同时将正在使用的范围内文件移动到新的条带组，然后通知客户端刷新并恢复 I/O。

当所有文件都成功迁移后，可将源条带组重新部署用于系统内的另一应用，也可停用并移除源条带组。

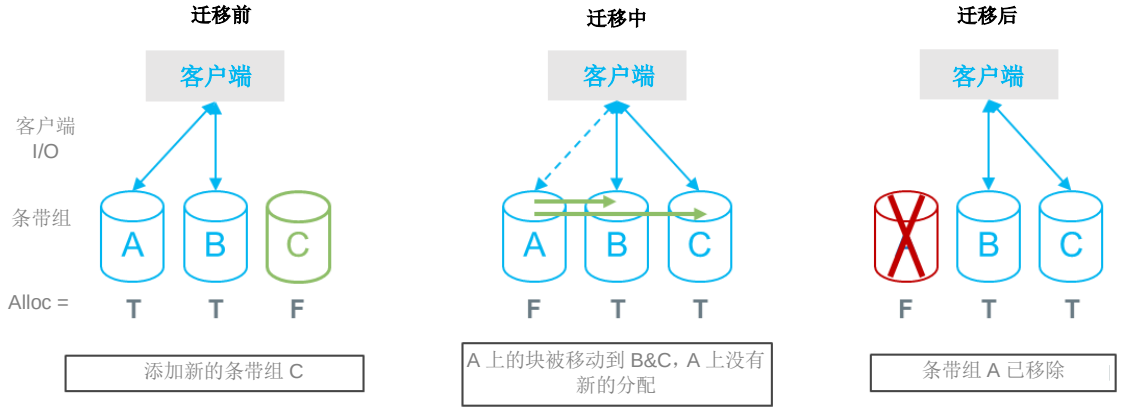


图 15-在线条带组迁移

配额

StorNext 系统通常会由多个部门中的多个用户共享，而每个部门都有不同的需求。在共享环境中，对系统资源进行管理，以避免冲突变得尤其重要。对于共享存储系统，通常会有两种资源需要分配：容量和性能。配额是用于分配容量的方法。两类存储资源配额可分配为：主存储和二级存储。可通过按需或定期管理报告的方式监测配额遵守情况。

主存储配额

主存储配额可用于管理主存储（例如 NVMe Flash、SSD 或 HDD）上的容量。文件系统配额有三种类型：用户配额、组配额和目录配额。用户和组配额可用于限制整个文件系统中用户或组可分配的容量。目录配额是用于限制分配给特定目录、及其子目录的容量，也可用于有效地限制文件数量。

每个配额可能具有与之相关联的两个限制：软限制和硬限制。硬限制是指：空间使用不应超过的绝对极限。每当总分配空间达到或超过硬限制时，违规用户、违规组或违规目录内的所有进一步写入请求都将被拒绝。

软限制是一个较小的限制。当应用空间超过软限制时，系统将启动可配置的宽限期计时器并发出警告。如果超过软限制的时间超出宽限期，软限制将转为硬限制并拒绝进一步写入。

为提供最大灵活性，可只指定硬限制，不指定软限制和宽限期，或者只指定软限制，有或没有宽限期。这种限制能够让管理员创建一系列从宽松到严格的实施方案。

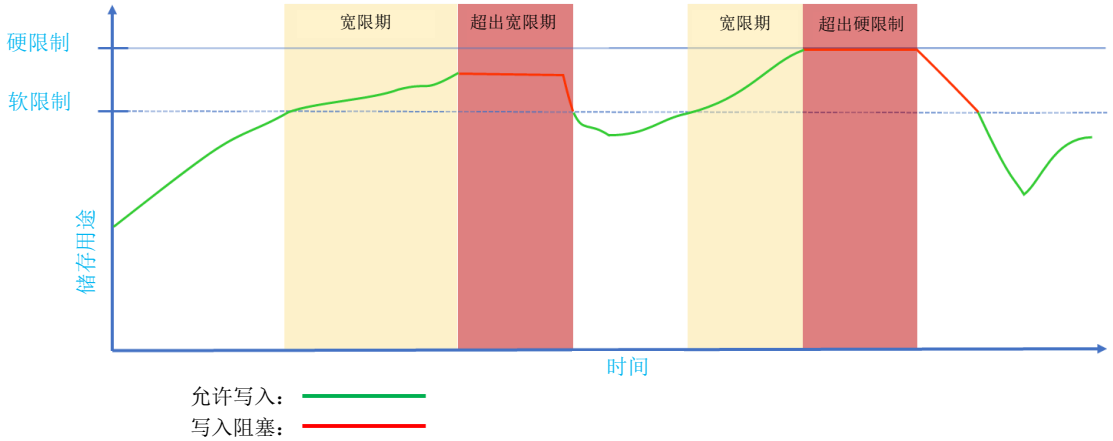


图 16-储存配额限制行为

二级存储配额

StorNext 具有一系列可管理次级存储的成熟而强大的功能，如下面的 FlexTier 部分所述。与主存储配额一样，二级存储配额为管理员提供了一种机制，通过该机制，可控制二级存储目标（如云、对象存储和磁带）的容量分配。此外，与主存储配额一样，二级存储配额涉及硬限制、软限制和宽限期的概念，并且被管理的实体包括用户和组。但二级存储配额涉及*项目*的概念，而不是简单的目录配额。项目是由管理员定义的，一个或多个目录及其子目录列表。这使得系统能够以不同的颗粒度级别，跟踪二级存储的使用。

二级存储配额也可以应用于特定的二级存储介质类型（例如：AWS）或媒体（例如特定的 AWS bucket），这有助于理解和控制“存储即服务”的使用。

服务质量/带宽管理（QBM）

分配 StorNext 系统的通量，是服务质量/带宽管理中包含的功能。不同客户端需要分配不同性能的原因有很多。一些可能需要特定级别的存储带宽，以执行其日常工作。还可能存在需在规定时间内优先于其他所有用户的特殊事件，例如新闻发布会或高管遴选。对于具有多用户浏览内容的系统，可能需通过一种方法保护内容创建者带宽不受观看者需求激增的影响。当然，StorNext QBM 可以处理的情况，远不止上述这些。

QBM 是基于条带组和客户端来配置的。定义每个条带组的总带宽容量后，根据客户端所属的类别，为客户端分配带宽。这里出现了四个类的定义，每个类的定义对应着不同的行为。带宽分配是动态变化的，并会根据存储的使用方式不断改变。总的目标是确保客户端能够使用所有可用的存储带宽（即，如果没必要，则不会对其进行限制），但是必须确保较高优先级的客户端始终可获得其需要的带宽。

QBM 配置的构建块，除服务类别外，还可将每个客户端配置为具有最小和最大带宽的分配。其目的是为了通过可用的总带宽、类和请求的带宽的组合，确定将分配给每个客户端的带宽。

类	优先级	说明
先到先得	第一	客户端至少收到请求的最小带宽，或者没有接收到最小带宽，但将在添加新客户端时保留其带宽。如果新的先到先得客户端由于可用带宽短缺而被拒绝，则将该客户端分配至公平共享类别。
公平共享	第二	客户端可与其配置成按照一定比例的共享类带宽。随着客户端的增加，每个客户端可用的带宽减少。
共享程度低	第三	这些客户端可共享，且带宽未被较高优先级类占用。适用于可在资源可用时工作的后台任务。
Mover	第四	用于限制 FlexTier 数据移动客户端的特殊类别，确保数据向次级存储的后台移动时，不会影响其他客户端。

表 4-QBM 服务类

FlexTier

StorNext FlexTier 是支持将 StorNext 文件系统扩展到次级存储子系统的通用名称。使用 FlexTier 特性的 SNFS 称为托管文件系统。“将 SNFS 扩展到次级存储”是指使用策略系统，根据需要在主存储介质和辅助存储最终介质间完成数据移动，从而使主存储池看起来比实际大很多。数据移动发生在后台，对用户和应用程序来说是不可见的（表面上）。无论数据块实际位于何处、或是否存在多个副本，文件始终出现在写入它们的文件系统中。以这种方式管理文件系统可以节省成本，因为旧数据可以一种透明的方式转移到更便宜的存储介质上，从而减少了对昂贵的主存储容量的需求。这是任何分级存储管理（HSM）系统的基本价值前提。

但是 FlexTier 不仅仅拥有简单的 HSM 功能。由于 FlexTier 可创建副本并保留文件版本，因此可使 StorNext 系统处于完全的自我保护中，并且无需额外的备份软件。文件在创建和更改时会始终受到保护。

二级存储的最终介质

二级存储的一般特征在于，每 TB 的存储成本是低于主存储的，但通常性能也较低。StorNext 二级存储选项包括 ChangHong Scalar 磁带库和 ChangHong ActiveScale 对象存储，但并不仅限于 CH 硬件。StorNext 支持许多第三方的存储设备和公有云。有关当前兼容列表，请参阅 StorNext 兼容性指南。请记住，新设备和云服务通常根据客户请求添加的。如需申请新二级存储设备或服务的认证，请联系 CH 公司。

FlexTier 支持以下类型的次级存储：

技术	成本	性能	说明
磁盘	$$$	最快	将 NAS 或块存储用作“存储磁盘”（SDISK） 需经常重新使用的上一代主存储硬件
对象存储	$$	较快	本地对象存储
公有云	$$	快到非常慢	非本地公共云存储提供商，包括热存储层和冷存储层
磁带	$	大文件顺序读写快 首字节延迟	使用 LTO 和企业驱动器的自动化磁带库

表 5-FlexTier 支持的次级存储类型

生命周期流程

在托管文件系统中，文件数据会根据管理员定义的策略移动。本小节将描述一套较为简化的数据基本生命周期流程，下一节还将介绍一些可用的策略选项。当然，用户和应用程序可将 StorNext 文件系统视为一个单独的命名空间，其可以像其他文件系统一样，用于存储和检索文件，而 FlexTier 活动通常是不可见的。当有时用户和应用程序需要看到“幕后”，StorNext 提供了诸如“脱机文件管理器”等的工具，可用于展示 FlexTier 的操作细节。

生命周期流程的策略，通常是应用于文件系统中的目录级。单个策略可应用于多个目录，并且并不要求每个目录都具有与其相关联的策略。

当文件被创建或复制到 SNFS 中时，其基本生命周期流程就开始了。系统会监视文件的状态，并且当它在几分钟内保持不变时，按照策略指示，会将文件的一个或多个副本存储到次级存储的最终介质上。如果定义了两个副本，则现在系统内总共存在文件的三个实例--原始文本会被保留在主存储上，两个副本则被保留在次级存储上。由于其他副本会被存在于其他介质上，因此实现了对该文件的保护，避免由于主存储系统出现可能的恶性故障而导致的数据丢失，因为于该文件已经“备份”了。如果两个副本中，至少有一个并不保存在本地（例如，保存在公有云存储中），那么也同样是对该文件实现了容灾保护，从而免受站点灾难的影响。

如果在存储后该文件被访问且有更改的情况发生时，更改后的文件副本，也将在静置几分钟后被存储到次级存储。这将被创建为文件的另一种版本。副本可从灾难性硬件故障中恢复，而文件版本可从更常见的事件（如意外遭更改）中恢复。“垃圾桶”功能可用于防止意外删除。

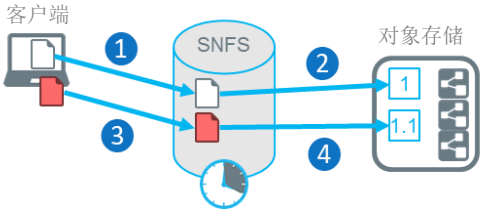
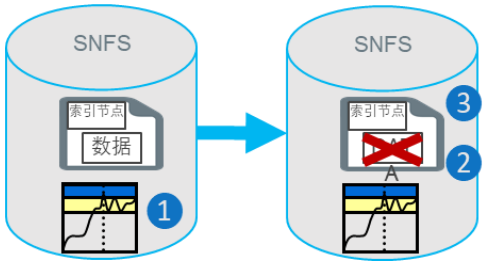


图 17-存储、副本、版本过程

序号	操作
1	存储文件
2	无活动计时器超时，创建副本
3	更改文件
4	计时器超时，创建版本

当主存储容量开始被逐渐占据时，我们会看到一个很有意思的现象。当达到被占容量较高时，*截断*会开始起作用。截断是指释放主存储空间的一个过程，其主要原理是，移除那些与先前已存储到次级存储中文件相关联的数据块，实现对主存储上的容量释放。文件的索引节点会被保留在主存储中，因此客户端能看到的文件列表不会有任何变化，但截断文件的数据块将只驻留在次级存储中。截断会一直运行，直到主存储容量消耗，降低到低水平以下为止。

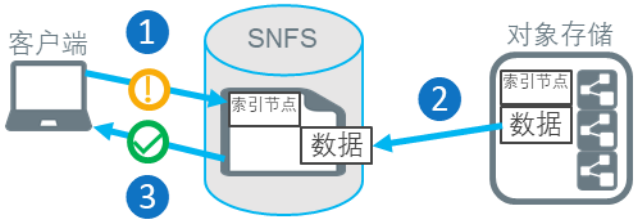


序号	操作
1	达到容量高水平
2	删除旧文件的数据块，直到达到低水平
3	索引节点仍然存在，文件对客户端可见

图 18-截断过程

截断功能会确保最近使用的文件是保留在主存储中的，而较少使用的文件，只保留在成本较低的次级存储中。截断的另一面被称之为*检索*。

当系统试图访问一个被截断的文件时，FlexTier 会从次级存储中检索文件的数据块，并将其写回到主存储中，然后再激活文件对请求者的可用。接着将检索到的文件保留在主存储中，直到其再次成为截断功能的候选文件。



序号	操作
1	客户端访问截断文件-访问暂时被阻止
2	从二级辅助存储器复制回磁盘的文件数据块
3	客户端被授予对文件的访问权限

图 19-截断文件的检索

生命周期选项

当你读到上一节的内容时，你可能有很多疑问。“我是否可调优该选项？”、“我是否可以改变此行为？”、“如果我不想使用该功能，是否可将其禁用？”由于 StorNext 是以客户导向为基础设计的，所以这些问题的答案也都是肯定的。几乎所有 StorNext 命令操作均可根据需求进行定制。表 7 总结了其中一些适用于所描述生命周期流程的重要选项。欲了解详细信息，请咨询在线 StorNext 文档中心。

功能	政策选项
存储	根据用户定义的计划或按需自动运行存储 根据保存的路径和文件名模式排除存储文件 在存储副本之前指定文件的创建历史 延迟存储直到数据量最少，或最长时限期满 存储文件时生成校验和
副本	副本份数（1-4 份） 每个副本的次级辅助存储的最终介质 副本过期（每个副本可能不同）
版本	要保留的文件版本数 删除前，保持非活动文件版本的时间长度
截断	文件将被截断之前的最短保存时间 在文件存储后立即截断它们 将特定文件“固定”到主存储，这样它们就不会被截断 根据保存的路径和文件名模式将文件排除在截断列表外 除了索引节点之外，在主存储上保留存根文件（大小可配置）
检索	检索顺序--当存在多个副本时（例如，首先是本地对象存储，其次是云存储） 检索文件时验证校验和 手动检索文件（对于即将需要的预暂存文件，手动检索非常有用。）

表 7--部分生命周期策略选项

离线保管库（Vaulting）

借用计算机科学家 Andrew S. Tanenbaum 向多伦多计算机服务大学主任 Warren Jackson 博士（大约 1985 年）的解释：“永远不要低估装满磁带的卡车的带宽。”虽然不像以前那么流行，但是在短时间内移动大量数据的最经济有效的方法仍然是磁带。利用 FlexTier 中的保管库（Vaulting）功能，可将拥有特定副本编号的所有磁带（例如：副本 2）移出自动磁带库。

保管库提供了一种将大型档案的副本安全转移到异地的快速有效的方法。除副本编号外，还有许多选项可用于自定义保管的内容、保管时间以及配置通知和报告，从而实现跟踪保管库的活动。保管库与导出（稍后讨论）的不同之处在于 FlexTier 保留了对保管介质的认知和所有权。例如，如果文件的本地版本被损坏或不可恢复时，FlexTier 将通知管理员将保管库的副本返回到系统以检索文件。出于安全性考虑，保管库还可对磁带实施加密保护，并且 WORM 介质甚至可用于确保数据在系统外时不能被篡改。

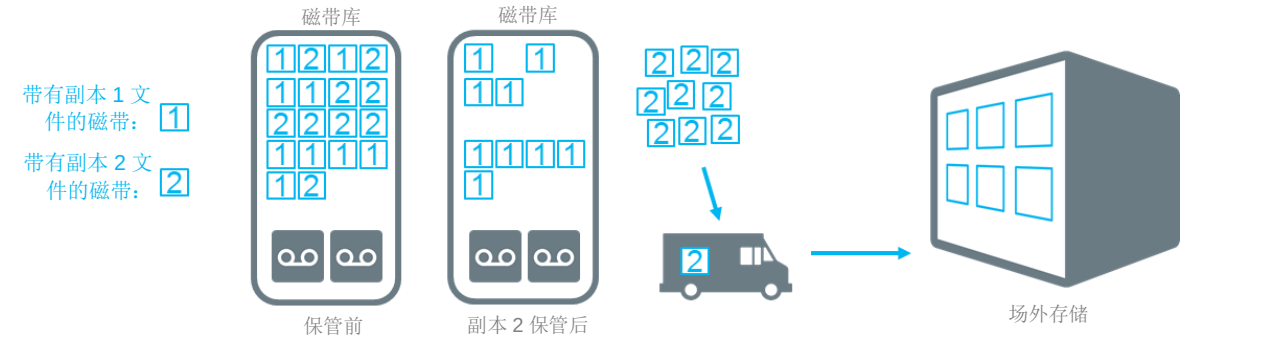


图 20–保管库流程

保管库还可以把包含几乎不被使用数据的磁带，从自动磁带库中移出，从而为最近的数据释放槽位空间。在这种情况下，磁带被移到其他空间，导致所需磁带库空间的缩小。

为了节省槽位而将磁带移出库的行为，存在一些弊端：手动处理磁带会招致因掉落、溢出以及灰尘和其他危险而造成的磁带损坏风险。但一个较为理想的选择方案在于，StorNext 的保管库功能完全被集成在了昆腾的 Scalar 库中，称为主动保管（Active Vault ，AV）。AV 可启用库内保管库，其中，保管库中的磁带将被移动至应用程序无法访问的单独保管库分区。AV 可使用未激活的槽位，相较于那些已经激活的槽位，这种槽位的成本更低。

使用者可将 StorNext 主动保管策略配置为：基于分区容量的填充水平，将候选磁带从 StorNext 分区自动移至 AV 分区中。许多参数可用于选择候选对象，包括媒体填充或空闲百分比，以及距离上次文件访问的时间间隔。当 StorNext 需访问驻留在主动保管库中的磁带时，管理员会收到警报，并可通过简单的 GUI 操作轻松将请求的磁带返回到 StorNext 分区。

AV 分区中的磁带无需离开库体，因此不会产生物理损坏的可能。由于它们是被存储在应用程序无法访问的槽位中（本例中展示的 StorNext 系统），因此它们是处在一种“与世隔绝”的环境里的，这同时可以保证文件防止受到勒索软件的攻击。有关使用主动保管策略，保护数据免受勒索软件侵害的详细信息，请参阅 Quantum.com 上的本技术简介。

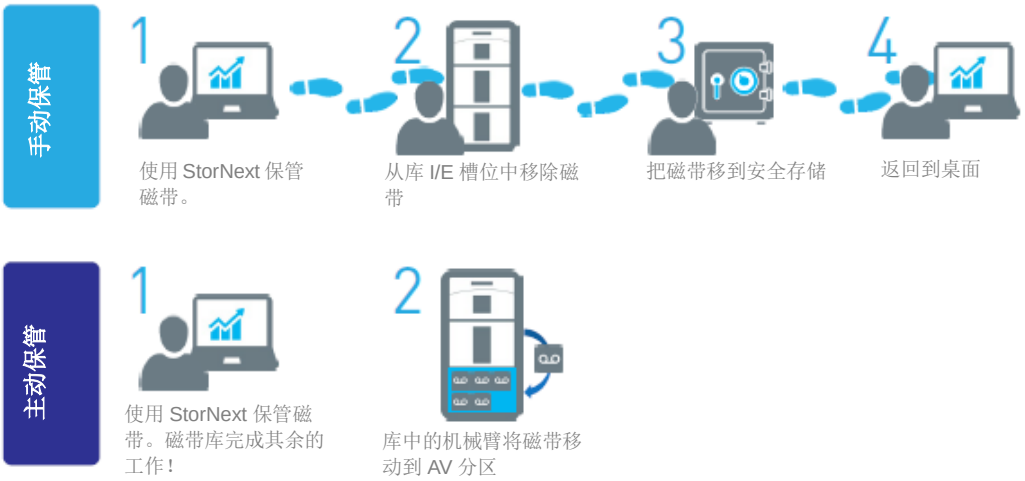


图 21-主动保管与手动保管

脱机文件管理器

文档中已经多次提到，默认条件下，FlexTier 活动对于 StorNext 用户和应用程序通常是不可见的。然而，不可见性可能是一把双刃剑，特别是在涉及云存储或磁带的情况下。根据文件的大小和所使用的存储层，可能会出现使用者在单击打开文件的几分钟或几小时后，文件才会变得再次可用并被移回磁盘。如果用户未注意到，文件其实是被保存在了较慢的介质上，他们可能会感到困惑和沮丧。在此类情况下，确保最终用户可以看到 FlexTier 的操作，且可对该操作的进行有限的控制，是至关重要的。FlexTier 脱机文件管理器（Offline File Manager ，OFM）是用于提供该类型可见性和控制的工具。

OFM 是一个基于客户端的工具，可用于运行在 Windows 和 macOS 系统中。它的功能是扩展现有的文件浏览器-Windows Explorer 或 Mac Finder。图标覆盖提供了一个快速的可视指示区，以说明文件仍在主存储上或已被截断；如果已被截断，则说明它是位于次级存储中了。上下文菜单可使用户能够按需存储、截断和检索文件，且无需管理员参与。

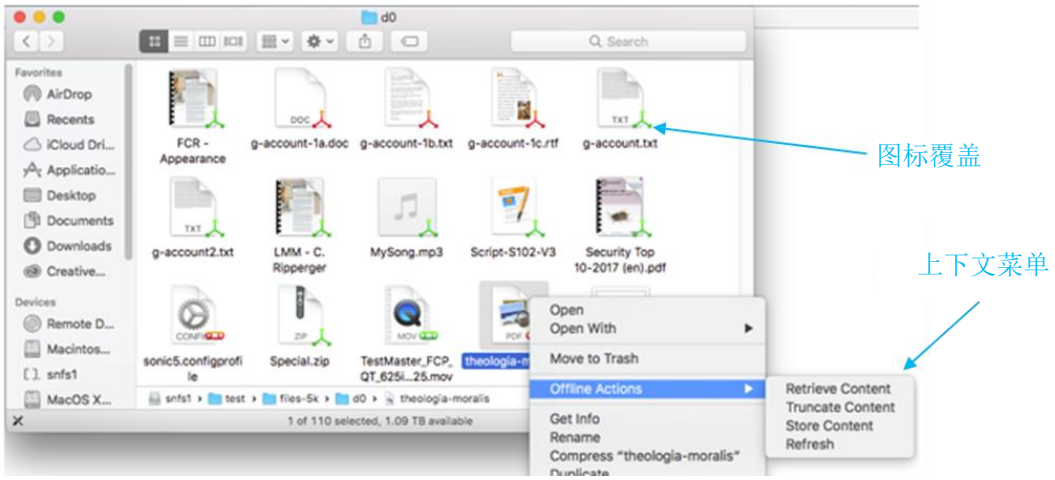


图 22-FlexTier 脱机文件管理器文件浏览器集成（Mac）

符号	说明	符号	说明
	文件在主存储上 文件在多个媒体上有副本		文件不在主存储上（被截断） 文件在多个介质上有副本
	文件在主存储上 文件副本被保存在在对象存储中		文件不在主存储上（被截断） 文件副本被保存在对象存储或云中
	文件在主存储上 文件副本被保存在磁带上		文件不在主存储上（被截断） 文件副本被保存在磁带上
	文件在主存储上 文件副本被保存在磁盘上		文件不在主存储上（被截断） 文件副本被保存在在磁盘上
	文件被保存在主存储上		文件不在主存储上（被截断）

表 8-FlexTier 脱机文件管理器图标展示

除文件浏览器集成外，还提供了客户端的 CLI，因此使用者也可以选择自主编写动作脚本。例如，如果用户需要定期在两个大型项目之间切换，则可通过编写脚本存储和截断项目 A 目录中的所有文件，并将项目 B 的所有数据从云端空间带回到主存储中。

FlexSync

FlexSync 是一套可简单、快速、高效地在本地或远程创建文件系统数据和元数据副本的工具，用于保护由 StorNext 管理的数据。同时它还是唯一可保护所有 StorNext 特定元数据（如 Windows ACL 和命名流），并完成复制和同步的工具，可提供完全一对一的副本。

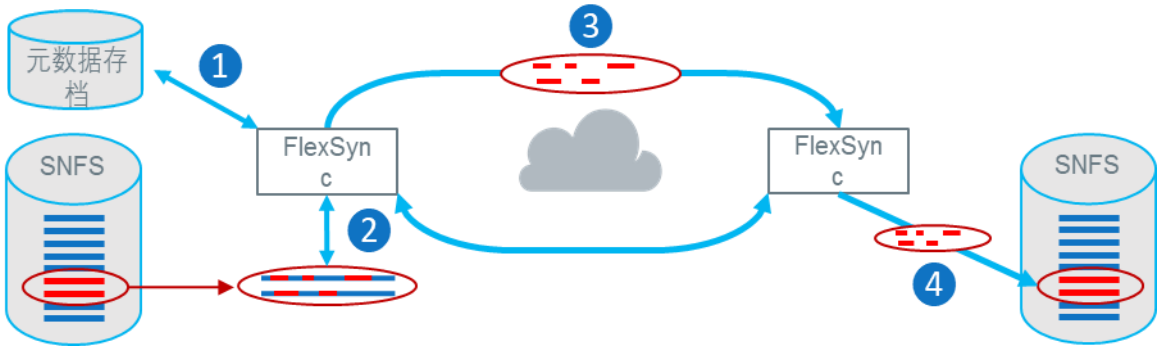
FlexSync 可用于复制单个目录、整个文件系统或其间的任何内容。典型的用途包括复制整个文件系统以进行 DR 保护，以及为进行内容分发，将一个或多个目录复制到多个远程站点。

FlexSync 的一些特性有助于其提升其性能，并巩固其的可扩展性。大多数复制工具，通常都是依赖于通过对文件系统扫描来检测更改，这种方法的效率低、速度慢且属于资源密集型的办法。对于那些大型文件系统，扫描可能需要几天的时间才能完成，这会对性能造成很大的影响。而 FlexSync 通过使用简单的数据库查询，利用 StorNext 的专利 MD 存档，检测更改，从而替代了整体扫描文件树的做法，避免占用过多资源的情况发生。

通过查询 MD 存档发现已更改文件后，系统会使用 delta 块压缩的技术，在文件源和最终介质间只发送更改文件的更改块-如图 23 所示。这使同步改变所需的网络带宽量极小。当更改到达时，文件即可在目标系统上实现完全重构。

FlexSync 可利用多线程和多流模式复制数据，甚至在单个任务中也是如此。为扩展复制能力，可在集群中的多个节点上部署 FlexSync 数据移动工具。

同步活动可为预设或手动启动，并且可通过仪表板的 GUI 轻松对其进行监控。



序号	操作
1	在元数据存档中查询源文件中的新文件或更改文件
2	比较块校验和，识别新的或改变的块
3	只向目标发送新的和更改的块
4	将更改内容合并至目标 SNFS

图 23-FlexSync Delta 块压缩

导入/导出

数据，尤其是最新的数据通常是不会静止不动的。顾名思义，“工作流”即代表相关的信息/数据/文件在类似于数字生产线流程中的移动。其过程可分解为：首先，获取源内容，然后对该内容进行各种形式的转换和修改，最终形成一个数字化产品，比如一部“超级英雄”电影、心脏肌肉的 3D 扫描图形，抑或是规划建筑中的虚拟游览景象。由于制作团队位于多个地点并且需要调用云端资源、文件迁移、第三方数据交换需求和等等其他因素，文件不仅需要在存储系统内完成移动，可能还需要被移入和移出存储系统。

StorNext 可通过各种导入/导出功能，解决这一问题。虽然在任何时候，文件系统均可将文件轻松从中复制到其他位置，然而这种模式只适用于少量的数据和偶尔使用的状况。如需定期移动大量数据，StorNext 的导入/导出功能就显得非常重要。

磁带

在磁带上导出文件有利于在 StorNext 实例间或 StorNext 与其他系统间移动大量数据。这在以下情况中是非常实用的：在远程站点间转移数据；制作副本供第三方（商业伙伴、监管机构、文化机构等）存档；将文件交付给客户，同时将其从 StorNext 中完全删除（这是在许多的行业合同中都会注明的）。

为实现在 StorNext 系统间传输文件，可使用格式为 ANTF 的磁带介质。ANTF 是 StorNext 的专有磁带格式，同时其在性能和效率方面进行了优化。为将文件传输到非 StorNext 系统，或当媒体的最终用户未知时，可按照行业标准 LTFS 格式导出媒体。对于任一介质格式，清单文件均可作为导出的一部分进行创建并用于后续的加速导入。还可以创建单独的报告文件，将其用作导出内容的可读记录，或作为创建批量导入文件的处理基础。

导出过程既允许导出选定源磁带介质上的所有文件，亦可导出单独指定、或需要批量处理的指定文件，该过程可导出和复制所有的原始数据。与所有 StorNext 功能一样，除 GUI 外，也可使用 CLI 命令，其有助于脚本的定制和自动化的实现。

为提供安全性和保护数据完整性，该过程支持磁带加密和使用 WORM 磁带介质。

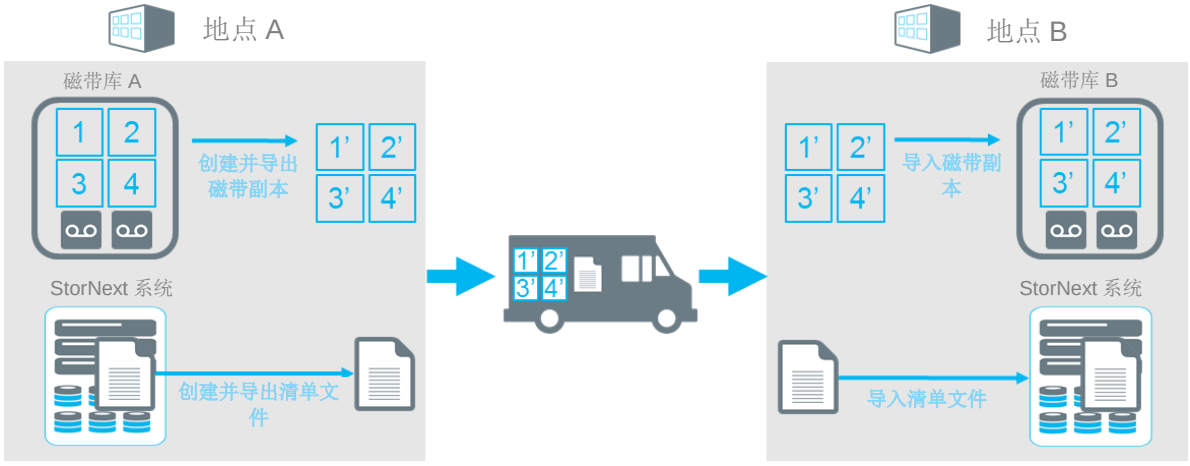


图 24-磁带副本导出/导入流程

通过导入过程，ANTF 或 LTFS 格式磁带上的文件能够快速导入另一个 StorNext 系统。与导出一样，可导入完整的介质，亦可只导入单独指定或批量处理文件中指定文件。导入操作可选用以下两种模式：*介质导入模式*和*文件导入模式*。

在介质导入模式下，磁带将被永久导入 StorNext，并通过磁带上的文件填充对应的数据库，然后利用对应于这些文件的截断文件填充目标目录。仅在请求文件数据块时，才从磁带中检索文件数据块。这将节省主文件系统存储上的空间并允许一定程度上的超额导入——导入的数据量可能比主存储上的可用空间还大。

文件导入会将文件从导入介质，复制到选定的目标目录，然后从系统中删除媒体。当文件导入完成后，须将磁带返回给发送方时，文件导入模式是非常有用的。

介质导入	文件导入
文件导入	在导入上创建的文件索引节点和元数据，以及复制到 SNFS 磁盘的数据块
在导入上创建的文件索引节点和元数据，以及复制到 SNFS 磁盘的数据块	磁盘上存在文件数据 - 未截断
文件访问时复制的数据块=检索延迟	无检索延迟，磁盘上已存在数据块
磁带须存放在库中	可将磁带从库中移除

表 9-媒体导入与文件导入

对象存储

如上文 FlexTier 部分所述，StorNext 能够向云存储和本地对象存储目标发送数据。但如果需将数据从对象存储转移到 StorNext 中时，该怎么办？这样做通常会涉及如下步骤和原因：

- 摄取云端中其他人共享的文件
- 在云存储中生成数据处理结果
- 将文件从老化的本地对象存储里，永久迁移到 StorNext 系统中
- 通过 StorNext，使现有对象存储中的内容在无需迁移的情况下实现可用
- 从云存储中过滤数据，确保实现高效、低成本的访问
- 其它原因。

对象导入可采用 FlexTier 中连接至对象存储系统的功能，但这其实是在引入数据，而非将数据发送出去。因为其使用了 FlexTier 的基础结构，所以对象导入也支持 FlexTier 种支持的任意对象存储目标。

与磁带导入一样，对象导入支持文件导入和介质导入两种模式。对于文件导入，源容器的内容将被复制到 SNFS 上的目标目录中。可通过一些选项，指定如何处理命名冲突。一旦导入完成，则可根据需要，将源容器与 StorNext 断开连接。

介质导入过程将首先扫描源容器并在 SNFS 中为找到的每个对象创建索引节点信息。此时所有对象均会被作为截断文件出现在文件系统中，并且可立即被 StorNext 客户端访问。与其他截断的文件一样，数据块也是按需调用的。当然，对象存储容器需保持连接才能工作。

此外，与磁带导入类似，用户可选择导入对象存储容器的全部内容，也可选择只导入指定对象。用户也可修改列表功能可并将其用于指导导入的对象列表。

对象不是文件，在此，我们需针对对象导入是如何将对象映射到文件系统命名空间，这一问题进行说明。对象存储容器是一个平面化的命名空间，而文件系统是包含文件夹和子文件夹的层次结构。对象可通过一个隐秘的一维名称（例如“FIOVSNERH34 6HSA0”）或一个看似是（但实质上并非）路径的名称（例如“/images/Alaska/bearsfishing.jpg”）存储在对象存储中。

如果导入的容器中的对象具有一维名称，则对象导入将在单个平面目标目录中，为每个对象创建一个文件，每个文件的名称与源对象名称相同。如果对象名称包含定义的分隔符，如正斜杠（“/”），则对象导入后，将创建为文件夹结构，该文件夹结构中包含有镜像对象名称中隐含的路径。对于上面的示例，导入过程将创建一个带有子文件夹/Alaska 的文件夹/图像，其中包含一个名为 bearsfishing.jpg 的文件。前斜杠是默认分隔符，但使用者也可指定使用其他任何字符。

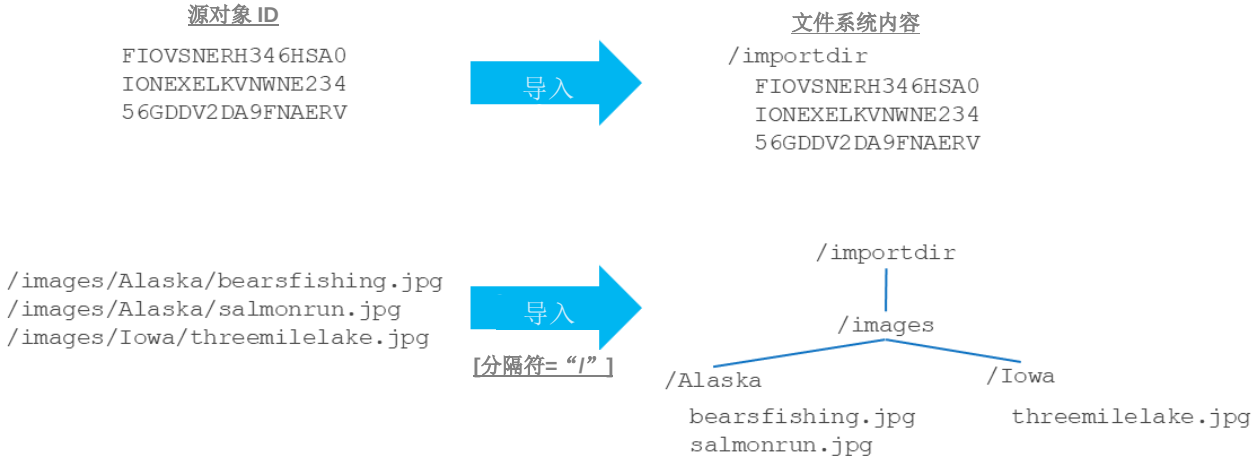


图 25-使用分隔符导入对象

导入过程支持两种导入筛选器，时间和前缀。时间筛选器，允许将导入限制为仅在指定时间之后创建的对象。因此，可实现对容器的“监控”，或定期扫描容器中导入的新内容。前缀筛选器，仅限于允许导入名称以指定前缀开头的对象。参见上面的示例，指定前缀“/images/Alaska/”将导入所有具有该前缀的对象，包括 bearsfishing.jpg。

转储

使用者考虑更改其归档软件的原因有很多：他们可能对当前产品的某些功能、或其支持形式、或定价规则不够满意，也可能只是因为，当前功能无法满足用户的需求，因此需找一些新的、可扩展性能更好的软件功能。从一个归档产品迁移到另一个归档产品过程面临的最大障碍，就是如何将数据从旧系统中转移到新

系统，同时保持对内容的持续访问。就像给行驶中的汽车换轮胎一样，要做到不受干扰是有些困难的。转录数百 TB 或 PB 的内容可能需数月或数年时间。在转换过程中，用户必同时使用两个系统，而管理层要为新老产品的许可和支持服务买单。

当然，用户还可以有更好的选择。如果 FlexTier 可以直接导入现有归档产品的数据库，且“不更换”保存内容所使用的介质，那会怎么样？通常这种转换将在瞬间结束也不需要冗长的转录过程。

这正是 StorNext 文档转换服务可提供的功能。原系统归档文件通过 StorNext 的 FlexTier 功能，可与 StorNext 的高性能存储体系架构进行集成时，保持数据的可访问性。

因为每个客户都有不同的需求和部署细节，所以文档转换工作需要定制化的专业服务来完成。即使无法采用介质来导入数据库，经验丰富的昆腾专业服务团队也可帮助客户通过更传统的遗留文档迁移支持 StorNext，从而最大限度减少不必要的麻烦。

Web Service API

每位客户对存储系统都有独特的需求，且会以特殊方式将其集成到自身的操作体系中。因此，StorNext 功能被定向为了几种不同的方式。GUI 和 CLI 适用于管理员发起的即时任务和配置。正常的后台进程，可使用内置的自动化和调度功能。CLI 可与自定义脚本相结合，使常规任务或事件序列自动化。为实现与不同应用的集成，昆腾提供了具有高度功能性的 Web Service API。

许多 CH 技术合作伙伴已开始通过 web 服务，将其的应用与 StorNext 进行集成。借助 web 服务，应用可获得 StorNext 内部状态的相关信息，诸如文件是否已被存储到二级辅助存储，以及具体存储位置，或替代默认的策略自动化，直接驱动 StorNext 完成操作。例如，当晚上 WAN 的使用率较低时，媒体资产管理器（MAM）可指示 StorNext 从云存储调用一组文件。用户只需与应用程序（在本例中是 MAM）简单交互，就可通过该应用程序，实现所需的 StorNext 系统内的任何操作。

系统在默认情况下，是禁用 Web 服务 API 访问的。启用时，可将其配置为使用 HTTP 或 HTTPS，并要求对请求进行身份验证（用户 ID 和密码）或不进行身份验证。系统内可创建多个 web 服务用户，每个用户均可拥有不同的访问权限。为了安全起见，Web 服务命令被分为四类。对于每个类别，每个 web 服务用户均具有读-写访问权、只读访问权或无访问权（禁用）。

安全类别	说明
文件控制	用于所有与文件相关的 Web 服务调用
最终介质控制	用于处理某特定介质中的所有 Web 服务调用
系统控制	用于所有与系统相关的 Web 服务调用
策略控制	用于所有与策略相关的 Web 服务调用

表 10-Web 服务 API 安全类别

各个 web 服务命令，能够收集大量的状态信息，并对系统进行全面的控制。从功能上讲，这些命令可以分为几类，总结在表 11 中。

功能类别	说明
归档	返回有关归档的信息、查询归档端口或更改归档状态
目录	修改目录的类属性或从媒体中检索或恢复文件
驱动	报告或更改驱动组件和存储子系统的状态
文件	报告、检索文件并将其存储到分层存储
作业	返回有关作业的信息
（介质）媒体	管理（介质）媒体-复制、清理、移动和报告
对象存储	关于对象存储组件的报告
策略	管理和报告策略
配额	管理和报告配额
报告	返回子系统资源请求信息
调度	管理和报告的时间表
系统	获取系统和 FlexTier 组件的状态。管理和报告备份。

表 11-Web 服务 API 命令功能类别

可在 StorNext 文档中心查阅 StorNext Web 服务 API 的各项综合指南。该指南包括所有可用功能和选项的详细信息、入门技巧、Java、Perl 和 Python 中的示例代码以及故障排除指南。

数据安全性的几句话

除上文已提到的特性（AD/LDAP 集成、SED、ACL、跨平台权限、校验和、加密、WORM）外， CH 还可通过其他方式确保运行 StorNext 的客户数据安全。

StorNext 软件和 StorNext 设备的升级，具有在发行过程中计算出的 MD5 校验和。该校验和是公开的，并且很容易通过公开可用的工具进行验证，确保下载的软件和设备及相关更新的完整性。对于由 CH 原厂和服务合作伙伴执行的升级，验证校验和是任何升级操作的标准部分。

StorNext 发行版在正式发布之前，已通过多个商业漏洞扫描程序进行测试。 CH 会对发现的任何问题进行研究，如有必要，公司会在发布前进行对应的代码修复。 CH 还将与执行自己漏洞扫描的客户合作，协助他们对出现的任何“被攻击”行为，进行解释和分析。

会持续关注 CISA 和其他 CSIRT 发布的漏洞数据库，评估新发现的安全问题并在必要时进行补救。主动向客户和合作伙伴通知关键问题和修复措施。

如果需根据 CH 的 TISAX 建议进行部署，则 StorNext 已通过毕马威的专业评估并公告为“TISAX 准备就绪”。这意味着相关需要 TISAX 合规性证明的组织可使用 StorNext。

数据安全是 CH 的一个重要课题。StorNext 的路线图中，包含了一系列新的，与其他数据安全相关的增强功能，我们很愿意与用户在 NDA（保密协议）框架下进行相关讨论。

结论

无论业界对非结构化文件存储的需求如何，StorNext 存储均可完美满足该类型的需求。出于其自身的卓越性能、可扩展性和灵活性的独特组合，StorNext 成为了那些致力于应对海量数据（这些数据须可用、可共享、受保护且存储成本低）的组织的理想选择。

关于CH

CH的技术与服务可帮助客户生成、编辑、共享数字内容，并以最低的成本，数十年如一日地保存并守护这些内容的安全。CH的产品平台可为高分辨率的视频、图像以及工业物联网等行业的数据提供最佳的处理性能；CH解决方案涵盖了数据生命周期的各个阶段，从高性能读取、实时协作和分析，到低成本存档等各个方面。每一天，全球知名的娱乐公司、体育俱乐部、研究组织、政府机构、企业和云服务提供商们，都在通过CH的产品与服务，让人们的生活变得更加精彩纷呈。想要了解更多，请访问www.changhongit.com.

www.changhongit.com.

WP00253A-2020 年