VSTECS UIH

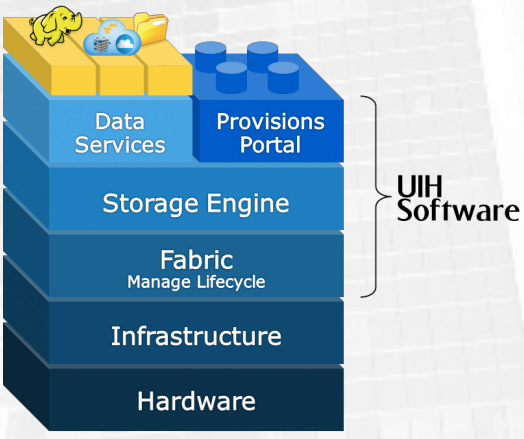
软件定义的云规模对象存储平台

產品概述

VSTECS UIH是一种软件定义的云规模对象存储平台，我公司在北京市海淀区设有独立的研发实验室，实验室共有研发人员30名。第一个版本于2012年发布。它兼具商用基础架构的成本优势以及传统阵列的可靠性、可用性和可维护性。借助 UIH，任何组织均可提供可扩展的简单公共云服务，同时拥有私有云基础架构的可靠性和控制力。 UIH 在单一的云规模存储平台上为非结构化（对象和文件）工作负载提供全面的协议支持。您可在单个全局命名空间下轻松管理全局分布式存储基础架构，并能随时随地访问内容。UIH 拥有灵活的软件定义的体系结构，该体系结构进行了分层，以促进实现无限的可扩展性。每层均可完全抽象、独立扩展、并具有高可用性且没有单点故障。

UIH的价值

- 云规模 — UIH 是适用于传统和新一代工作负载的对象存储平台。它拥有灵活的软件定义的体系结构，可促进实现无限的可扩展性。功能亮点：
  - -EB 级规模
  - -全局分布式对象基础架构
  - -在一个存储平台中支持数以十亿计的所有类型（大和小）的文件
- 灵活的部署 — UIH 具有无与伦比的灵活性，可部署为应用装置纯软件解决方案或部署在 EMC 运营的云中（或是同时采用两种形式）。提供的功能：
  - -应用装置部署（U 系列、D 系列）
  - -支持经鉴定的行业标准硬件的纯软件部署。
  - -多协议支持：对象、文件、Hadoop
  - -多个工作负载：物联网、归档、大数据、现代应用程序
  - -CIFS-UIH：免费附加软件可通过 Windows 文件系统接口实现 UIH 访问
  - -NFs 云层：一个目标用于所有主存储



- 企业级 — 通过在具有诸如以下这类功能的安全和兼容系统中采用企业级对象、文件和 HDFS 存储，UIH 可为客户提供对数据资产的增强控制。
  - -静态数据和跨站点复制加密
  - -针对 SEC 17-A4 法规遵从性的报告、基于策略和基于事件的记录保留和平台加强，包括高级保留管理（如诉讼封存和最小值-最大值治理）
  - -使用 Active directory/LDAP 实现的身份验证、授权和访问控制
  - -与监视和警报基础架构（SNMP 陷阱和系统日志）的集成
  - -空间回收
  - -增强的企业级功能（多租户、Swift 多部分上载、容量监视、警报等）

- 总体拥有成本 (TCO) 降低 — 相对于传统存储以及公共云存储，UIH 可以大幅降低 TCO。它甚至可为 LTR 提供比磁带更低的TCO。其特性包括：
  - -空间回收
  - -增强的企业级功能（多租户、Swift 多部分上载、容量监视、警报等）
  - -全局命名空间
  - -小型和大型文件性能
  - -无缝 Centera 迁移
  - -每个机架的原始容量从 3.8 PB 增加到 6.2 PB
  - -比公共云存储 (TCS) 更低的总体拥有成本 (TCO)
  - -比其他对象存储解决方案更多的 TCO
  - -管理开销低
  - -数据中心占用空间少
  - -存储利用率高

UIH 的设计针对以下主要使用情形进行了优化：

- 区域防护归档 — UIH 可作为用于归档和长期保留的安全且经济实惠的本地云。将 UIH 用作归档层可以显著减少主存储容量。在 UIH 2.2 和更高版本中，对冷归档使用了 10/2 擦除编码方案，其中区块会划分为 10 个数据片段和 2 个编码（奇偶校验）片段。这样可以为此特定使用情形实现更好的存储效率。
- 全局内容存储库 — 非结构化内容存储库（包含图像、视频等）当前存储在高成本存储系统中，这使企业无法经济高效地管理大规模数据增长。借助 UIH，任何组织均可将多个存储系统整合到一个可全局访问的高效内容存储库中。
- “物联网”存储 — 对于能够从客户数据中提取价值的企业，物联网提供了新的创收机会。UIH 提供了一种高效的“物联网” 体系结构，支持大规模的非结构化数据收集。由于可以直接在 UIH 平台上分析数据，而无需执行耗时的 ETL（提取、转换、加载）过程，因此 UIH 还可以简化分析。只需将 Hadoop 群集指向 UIH 即可开始运行查询。作为对象存储系统，UIH 可以采用非常经济高效的方式存储所有物联网数据，并利用其内置数据检查来确保数据完整性。
- 视频监控 — 与物联网数据相反，视频监控数据的对象存储计数要小得多，但是每个文件的容量占用空间要高得多。尽管数据真实性十分重要，不过数据保留并没有这么关键。UIH 可以是用于此类数据的低成本存放区域或辅助存储位置。视频管理软件可以利用 UIH 丰富的元数据，通过重要的详细信息（如摄像机位置、保留要求和数据保护要求）对文件进行标记。UIH 的元数据还可以用于将文件设置为只读状态，以确保对文件实现监管链。
- 现代应用程序 — UIH 提供了一种专为现代应用程序开发、管理和分析而设计的单一全包式平台。UIH 可支持新一代 Web、移动和云应用程序。其多站点读/写机制可确保高度一致性，从而大大简化了开发人员的工作。当 UIH 容量发生变化和增长时，开发人员从不需要重新编写其应用程序的代码。
- 数据湖基础 — UIH 可为任何规模的组织建立数据湖基础。它通过强大的 HDFS 服务最大限度利用用户数据，使得在生产中利用大数据应用程序和分析成为现实。它采用“原位”分析功能来减少风险和资源消耗并缩短了实现成效的时间。

体系结构

UIH 在构建时遵循了某些设计原则，例如具有高度一致性的全局命名空间、无限的容量和横向扩展、安全多租户以及适用于小型和大型对象的卓越性能。UIH 按照云应用程序的原则构建为完全分布式系统。在此模型中，每个硬件节点都与系统中的其他节点相同。在这些节点上运行的 UIH 软件形成了底层云存储，从而提供保护、异地复制和数据访问。本部分会深入介绍 UIH 体系结构以及软件和硬件设计。

概述

- UIH 提供一个软件定义的云存储平台，可以部署在一组经鉴定的行业标准硬件或全包式存储应用装置上。在高层次上，UIH 由以下主要组件组成：
- UIH 门户和调配服务 — 提供一个基于 Web 的门户，通过它可实现 UIH 节点的自助服务、自动化、报告和管理。它还处理许可、身份验证、多租户和调配服务。
- 数据服务 — 提供服务、工具和 API 来支持对象以及 HDFS 和 NFSv3。
- 存储引擎 — 负责存储和检索数据、管理事务以及保护和复制数据。
- 结构 — 提供群集、运行状况、软件和配置管理以及升级功能和警报。
- 基础架构 — 使用 SUSE Linux Enterprise Server 12 作为全包式应用装置的基本操作系统或行业标准硬件配置的经鉴定的Linux 操作系统。
- 硬件 — 提供全包式应用装置或经鉴定的行业标准硬件。

UIH 门户和调配服务

UIH 管理通过 UIH 门户和调配服务进行。UIH 提供基于 Web 的 GUI，使您可以对 UIH 节点进行管理、许可和调配。该门户具有全面的报告功能，其中包括：

- 每个站点、存储池、节点和磁盘的容量利用率
- 针对延迟、吞吐量、每秒事务处理数以及复制进度和速率的性能监视
- 诊断信息，如节点和磁盘恢复状态以及有关硬件和进程运行状况的每节点统计信息，这些信息有助于标识性能和系统瓶颈。

数据服务

可通过对象、HDFS 和 NFS v3 协议访问存储在 ECS 中的数据。通常情况下，ECS 提供多协议访问，这意味着通过一种协议接收的数据可以通过其他协议进行访问。例如，可以通过 S3 接收数据，并通过 NFSv3 或 HDFS 修改数据（反之亦然）。根据协议语义和协议设计方式的表示，对于这种多协议访问有一些例外情况。表 1 突出显示支持的对象 API 和协议以及进行互操作的协议。

数据服务（也称为头服务）负责接收客户端请求、提取所需信息以及将它传递给存储引擎以便进行进一步处理（例如读取、写入等）。在 2.2H1 和更高版本中，所有头服务都已合并为一个在基础架构层上运行以处理每种协议的进程（称为“dataheadsvc”），从而可降低整体内存消耗。此过程进一步封装在名为 object-main 的 Docker 容器中，后者在 UIH 系统中的每个节点上运行。本文档的“基础架构”部分会更加详细地介绍此主题。此外，要通过以上协议访问对象，需要打开特定防火墙端口。

协议		支持	互操作性
对象	S3	其他功能，如字节范围更新和丰富的 ACL	HDFS、NFS
	Atmos	版本 2.0	NFS（仅限基于路径的对象，不包括基于 ID 样式的对象）
	Swift	V2 API 以及 Swift 和 Keystone v3 身份验证	HDFS、NFS
	CAS	SDK v3.1.544 或更高版本	N/A
HDFS		Hadoop 2.7 兼容性	S3/NFS、Swift/NFS
NFS		NFSv3	S3/HDFS、Swift/HDFS、Atmos（仅限基于路径的对象，不包括基于 ID 样式的对象）

支持的数据服务

对象

对于对象访问，UIH 提供行业标准对象 API。UIH 支持 S3、Atmos、Swift 和 CAS API。除了 CAS 以外，对象或数据通过 GET、POST、PUT、DELETE 和 HEAD 的 HTTP 或 HTTPS 调用进行写入、检索、更新和删除。对于 CAS，使用标准 TCP 通信以及特定访问方法和调用。

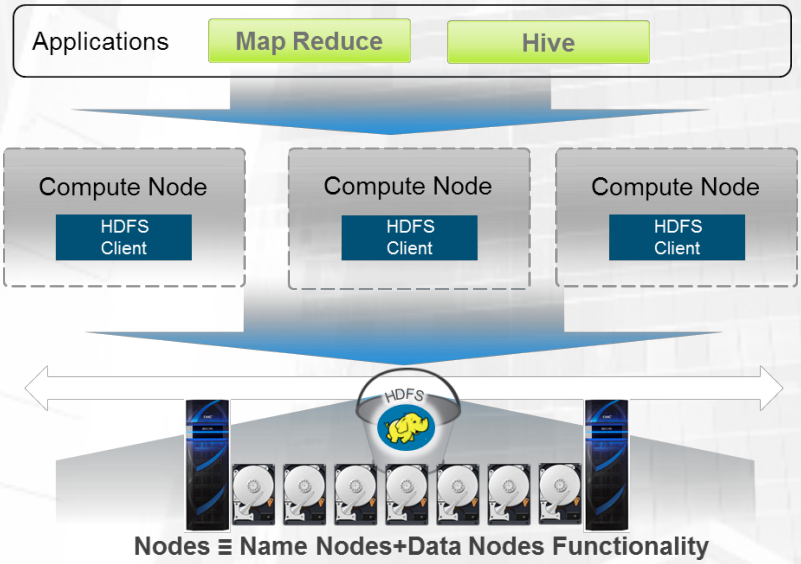
此外，UIH 为对象的元数据搜索提供了工具。这使 UIH 可根据存储区中对象的相关元数据维护这些对象的索引，从而使 S3 对象客户端可以根据已编制索引的元数据使用丰富的查询语言来搜索存储区中的对象。搜索索引可以是每个存储区最多 30 个元数据字段，通过 UIH 门户、UIH Management REST API 或 S3 REST API 在存储区创建时进行配置。

对于 CAS 对象， CAS 查询 API 提供类似功能来根据为 CAS 对象维护的元数据搜索对象，并且不需要显式启用。

HDFS

UIH 可以存储 Hadoop 文件系统数据，这使组织可以在 UIH 上创建 Hadoop 分析可使用和处理的大数据存储库。HDFS 数据服务与 Apache Hadoop 2.7 兼容，可支持细粒度 ACL 和扩展文件系统属性。

在 UIH 2.2 和更高版本中，UIH 已与 Ambari 集成，这使您可以轻松部署 UIH HDFS 客户端（jar 文件）并指定 UIH HDFS 作为 Hadoop 群集中的默认文件系统。如图 4 所示，在 Hadoop 群集中每个节点上都安装了 UIH HDFS 客户端文件。UIH 提供与 Hadoop 部署中的名称节点和数据节点所执行的功能等效的文件系统和存储功能。UIH 无需将数据迁移到本地 Hadoop 直连存储 (DAS) 和/或创建至少三个拷贝，从而简化了使用 DAS 的 Hadoop 的工作流。



UIH 2.2 中针对 HDFS 添加的其他增强功能包括：

- 代理用户身份验证 — 用于 Hive、HBase 和 Oozie 的模拟。
- 安全性 — 服务器端 ACL 强制实施与 Hadoop 超级用户和超级用户组的添加，以及存储区上的默认组。

UIH 2.2 使用 Hortonworks (HDP 2.3) 进行了验证和测试。它还为诸如 YARN、MapReduce、Pig、Hive/Hiveserver2、HBase、Zookeeper、Flume、Spark 和 Sqoop 这类服务提供支持。

文件

UIH 2.2.1 和更高版本包含使用 NFSv3 的本机文件支持。NFSv3 文件数据服务的主要功能包括：

- 丰富的 ACL — 支持丰富的访问控制列表（允许实现一组更复杂的权限模式和扩展属性的 ACL）
- 全局命名空间 — 能够从任何站点上的任何节点访问文件（使用负载平衡器）。
- 全局锁定 — 能够从任何站点上的任何节点锁定文件（共享和独占锁定、基于范围的锁定和强制锁定）
- 多协议访问 — 能够访问对象（S3 和 Swift）、HDFS 和 NFS 创建的数据。

NFSv3 通过 UIH 门户进行配置。NFS 导出、权限和用户组映射通过 API 和/或门户进行创建。此外，NFS 客户端（如 Solaris、Linux 和 Windows）可以使用命名空间和存储区名称装载在 UIH 门户中指定的导出。例如：mount –t nfs –o vers=3 <节点的 IP 或 DNS 名称>:<导出路径（即 /命名空间/存储区）>。要在节点故障期间实现客户端透明度，必须使用负载平衡。

UIH 紧密集成了其他 NFS 服务器实施（如 lockmgr、statd、nfsd 和 mountd），因而这些服务不依赖于基础架构层（主机操作系统）进行管理。

NFS v3 支持具有以下功能：

- 对文件或目录数没有设计限制
- 文件写入大小可以高达 4 TB
- 能够使用单个全局命名空间/共享在 8 个站点间进行扩展
- 支持 Kerberos 和 AUTH_SYS 身份验证

NFS 文件服务处理来自客户端的 NFS 请求；但是，数据存储为对象（类似于对象数据服务）。NFS 文件句柄会映射到对象 ID。由于文件从根本上说是映射到对象，因此 NFS 具有与对象数据服务类似的功能，包括：

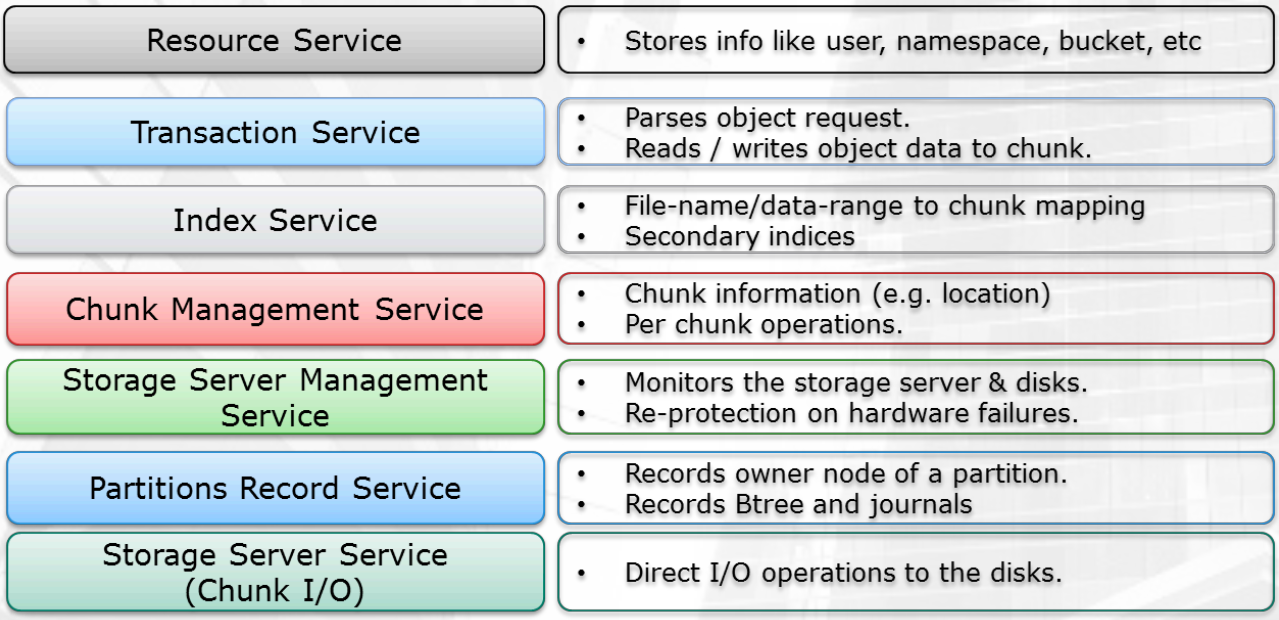
- 存储区级别的配额管理
- 对象级别的加密

存储引擎

系统的核心是存储引擎。此层包含负责处理请求以及存储、检索、保护和复制数据的主要组件。此部分介绍设计原则以及在内部表示和处理数据的方式。其中还介绍了读取和写入的数据流。

服务

具有分层体系结构，系统中的每个功能都构建为一个独立层。此设计原则使每层可跨系统中的所有节点横向扩展，并确保高可用性。UIH存储引擎包含下图所示的各层，这些层在基础架构和硬件组件上运行。

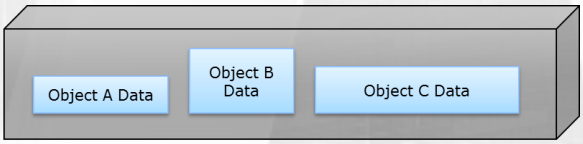


数据和元数据

存储在 UIH 中的数据的详细信息可以概括如下：

- 数据 — 需要存储的实际内容（图像、文档、文本等）
- 系统元数据 — 与数据相关的信息和属性。系统元数据可以分类如下：
  - -o标识符和描述符 — 一组在内部用于标识对象及其版本的属性。标识符是不在 UIH 软件上下文外部使用的数字 ID 或哈希值。描述符定义诸如内容和编码的类型这类信息。
  - -o采用加密格式的加密密钥 — 对象加密密钥在通过 KEK（密钥加密密钥）进行加密之后包含在系统元数据中
  - -o内部标记 — 一组标记，用于跟踪是否启用了字节范围更新或加密，以及用于协调对象锁定、缓存和删除
  - -o 位置信息 — 一组具有索引和数据位置信息（如字节偏移量）的属性。
  - -o 时间戳 — 一组用于跟踪对象创建、修改和到期时间的属性。
  - -o 配置/租户信息 — 对象名称、命名空间名称、分区/vdc 名称和对象的访问控制信息。
- 自定义用户元数据 — 用户定义的元数据，提供所存储的数据的进一步信息或分类。自定义元数据格式化为随写入请求一起发送的键/值对。

所有类型的数据（包括系统和自定义元数据）都存储在“区块”中。区块是连续空间的 128 MB 逻辑容器。请注意，每个区块可以具有来自不同对象的数据，如图 6 所示。UIH 使用索引跟踪可能在不同区块和节点间分布的对象的所有部分。区块采用仅附加模式进行写入，这意味着，应用程序无法修改/删除区块中的现有数据，而更新的数据在新区块中写入。因此，I/O 无需锁定，并且无需让缓存失效。仅附加设计还简化了数据版本控制。旧版本的数据在以前的区块中进行维护。如果启用 S3 版本控制，并且需要较旧版本的数据，则可以使用 S3 REST API 检索数据或恢复为以前的版本。

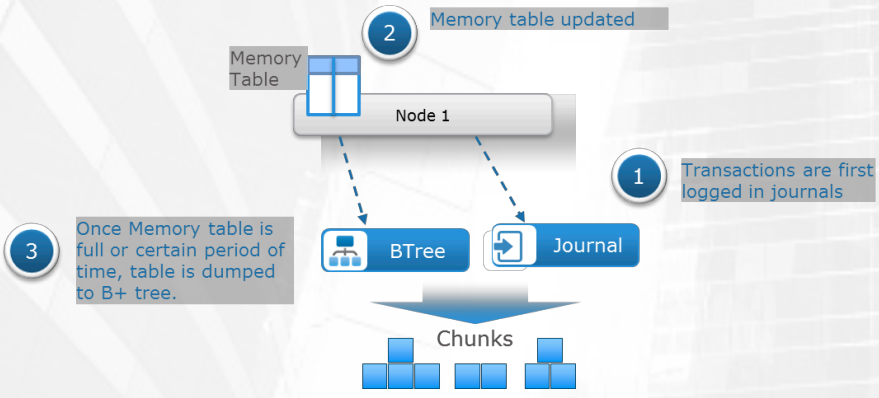


Chunk = 128 MB unit

所有区块都三重镜像到多个节点，以针对驱动器或节点故障保护数据。当纯数据块填充达到 128 MB 和/或进行密封时，数据块会进行擦除编码以提高存储效率，并跨多个节点分布以实现数据保护。区块进行擦除编码并受到跨节点保护之后，会丢弃镜像拷贝。包含索引和元数据的区块会进行三重镜像，但不进行擦除编码。本文档的“数据保护”部分中会提供三重镜像和擦除编码的更详细描述。

数据管理（索引和分区记录）

使用逻辑表跟踪数据和元数据区块在磁盘上的存储位置。这些表包含用于存储与对象相关的信息的键/值对。一个哈希函数用于快速查找与键关联的值。这些键/值对最终存储在 B+ 树中，以便对数据位置进行快速索引。通过将键/值对存储在平衡的搜索树（如 B+ 树）中，可以快速访问数据和元数据的位置。此外，为了进一步增强这些逻辑表的查询性能，实施了一个两级日志结构合并 (LSM) 树。因而有两个树状结构，其中较小的树处于内存中（内存表），而主 B+ 树驻留在磁盘上。因此，键/值对的查找会首先在内存中进行查找。如果值不在内存中，则会查看磁盘上的主 B+ 树。这些逻辑表中的条目首先记录在日志记录中，这些日志会作为区块写入磁盘并进行三重镜像。日志会跟踪尚未提交到 B+ 树的索引事务。事务记录到日志中之后，内存表会进行更新。一旦内存中的表已满或在一段特定时间之后，表最终会进行合并排序或转储到磁盘上的 B+ 树。下图展示了此过程



佳杰科技(上海)有限公司　网 址：www.ecschina.com
地 址：北京市海淀区长春桥路11号万柳亿城大厦A座6/7层 100089
电 话：010--58815599转支线6067　传 真：010–58818878转265