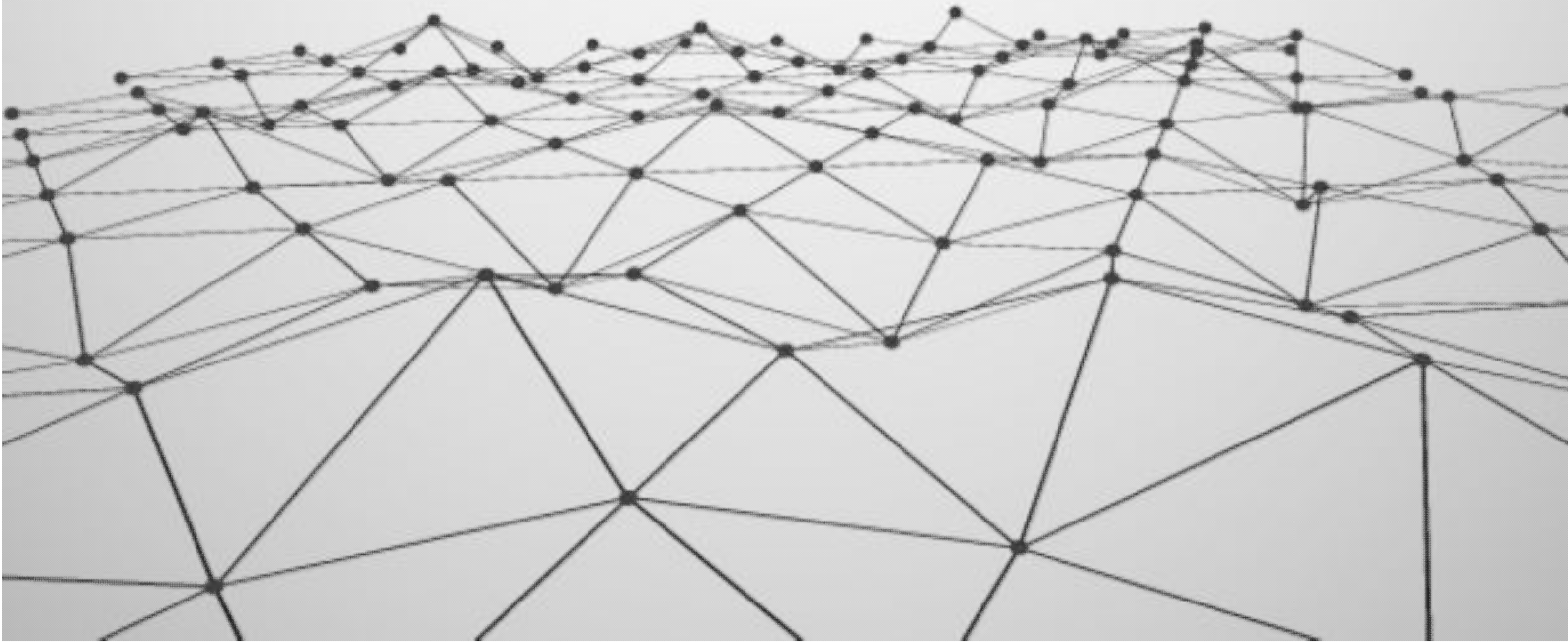
有容云
Yourun Cloud

VeSpace Server SAN 产品

技术白皮书

文档版本：V2.1 发布日期：2017-8-4



目 录

版权所有 ©深圳市云舒网络技术有限公司

版权所有 ©深圳市云舒网络技术有限公司

版权所有 ©深圳市云舒网络技术有限公司

一、VeSpace Server SAN 系统

1.1 系统简介

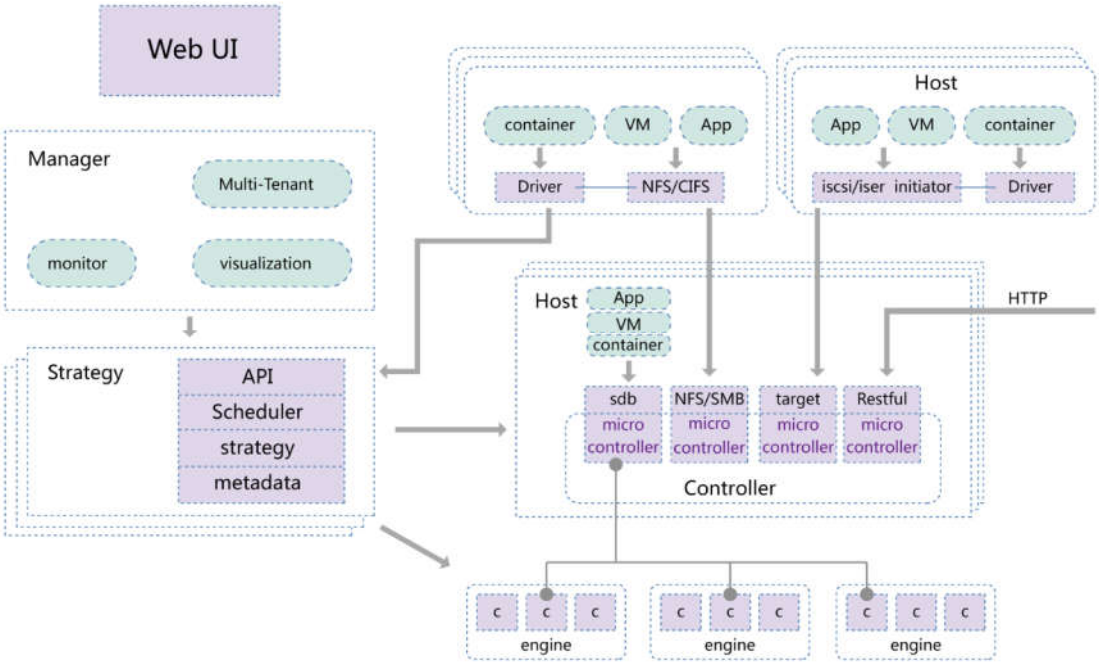
VeSpace Server SAN 是云舒网络自主研发的软件定义存储产品，是一款分布式存储软件，将通用 x86 服务器上的存储设备(HDD、SSD)抽象成存储资源池对外提供块存储功能。VeSpace Server SAN 专门为云计算生态而设计，在产品易用性、运营成本，数据安全和软件复杂度上进行了深入的思考与权衡，融合了分布式数据、分布式缓存、容量负载均衡、及多重数据保护等诸多存储技术，能够满足金融、互联网、制造业、电信、证券、电力、石油等行业关键业务的需求，保证客户业务高效稳定运行及数据安全，提升业务的敏捷性与竞争力，最大程度上让客户减少在存储系统上的 TCO（Total Cost of Ownership）。

1.1.1 VeSpace Server SAN 特点

- **全分布式布局**：存储集群全分布式冗余架构，无单点故障。
- **微服务架构**：模块间以 RESTful API 通信，松耦合，易于第三方以 API 方式访问存储集群。
- **无硬件依赖**：通用 x86 服务器，标准 Linux 系统，即可运行。
- **标准块接口**：无需特殊配置即可实现计算与存储的融合。
- **高性能**：尽可能缩短 IO 路径，增加多种 Cache 保证存储的性能。
- **高可靠**：可配置成多副本或纠删码方式确保数据安全，同时可配置强一致性策略确保各个副本数据的一致性。
- **智能数据恢复**：仅需要重建已存在并且不一致的数据，忽略卷从未写入数据的位置，提高重建效率。
- **可扩展性**：横向扩展，平滑扩容。集群扩容时，在条件允许的情况下，并不会移动数据，减少数据移动对集群负载带来影响。之后由智能调度算法，完成数据均衡和负载均衡。
- **敏捷性**：秒级创删卷，秒级跨主机迁移卷，适应容器应用等场景。

版权所有 ©深圳市云舒网络技术有限公司

1.1.2 VeSpace Server SAN 架构



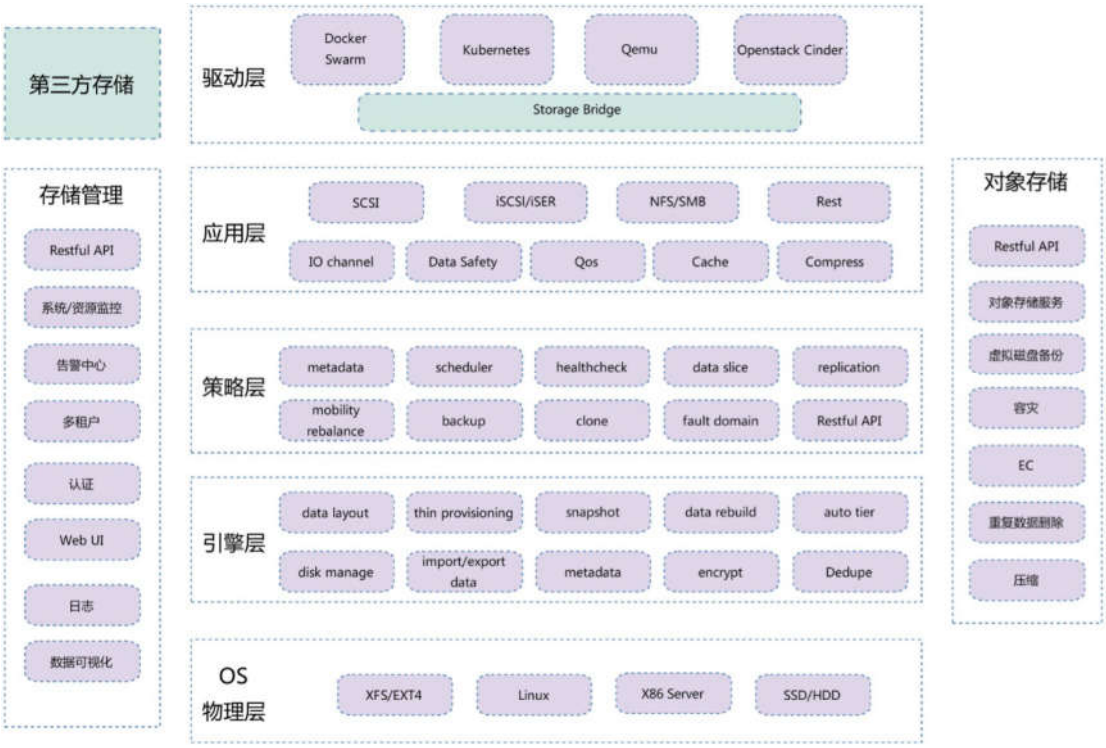
VeSpace Server SAN 划分如下模块：

- **Strategy**(策略子系统)
- **Driver**(驱动子系统)
- **Controller**(存储子系统控制层)
- **engine**(存储子系统引擎层)
- **Manager**(管理子系统)

所有子系统的详细特性，后续章节都有描述。

版权所有 ©深圳市云舒网络技术有限公司

1.1.3 VeSpace Server SAN 功能模块



1.1.4 VeSpace 未来

VeSpace Server SAN 已经成功发布，功能完善，并且运行稳定，后续会在性能，数据安全，高端存储特性等方面持续改进，确保其成为更优秀的产品。

如上图右边的属于 VeSpace 系列下一个产品 VeSpace Object Storage(VOS)。对象存储产品 VOS 实现异地部署、容灾、持续数据保护。VeSpace Server SAN 的卷可一键备份到 VOS 中，进一步保证数据安全。

1.2 应用场景

场景 1: 软件定义数据中心 IaaS

软件定义数据中心 IaaS 场景适合对存储需求量有较大的大中型的组织或者企业（比如金融、保险、互联网等）。软件定义数据中心 IaaS 场景是将通用的 x86 服务器资源池化，形成一个大规模的数据中心。大规模的资源存储池为标准的块设备存储提供了统一的访问接口。支持虚拟化 Hypervisor 平台和其他各平台的集成，比如 VeSpace Server SAN、KVM/QEMU 和 OpenStack 等。使用 VeSpace Server SAN 可以为软件定义数据中心 IaaS 获得更大范围的弹性调度能力，提高存储资源利用率、减少管理复杂度。

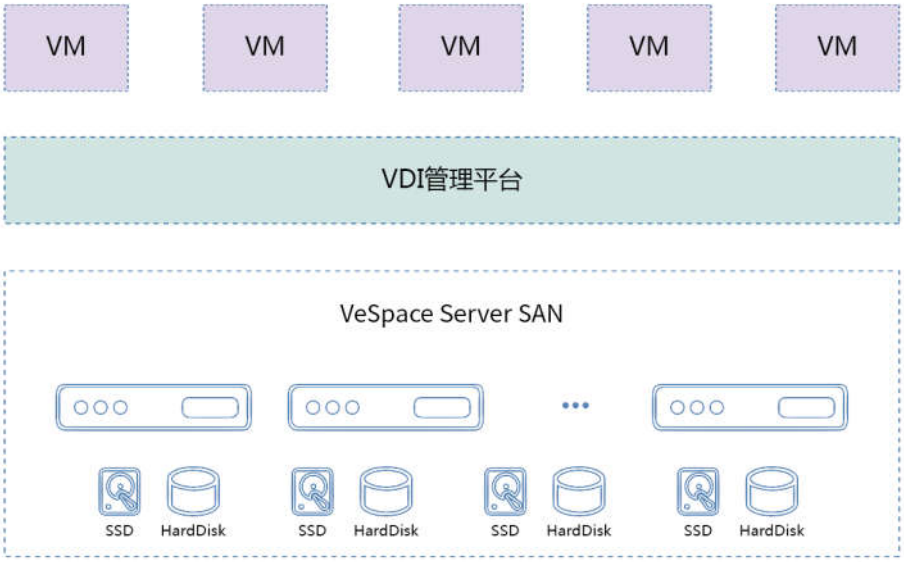
1.2.1 容器云是以容器为核心的云平台，容器云为传统的存储提出了三个挑战:

版权所有 ©深圳市云舒网络技术有限公司

1. 存储对容器快速响应: 容器在使用过程中需要频繁的创建和删除卷，类似这种操作在容器数量较多时，对后端存储的性能要求很高，需要后端存储快速响应，并帮助容器找到合适的后端存储资源。VeSpace Server SAN 提供了秒级创建和删除卷，快速响应容器。
2. 容器的迁移：容器在迁移时，需要卷信息和数据也要迁移到另外一台主机上，容器对存储的要求是数据一致性和持久化。VeSpace Server SAN 的迁移功能可以帮助容器高效、快速完成容器的迁移。
3. 应用感知：需要根据容器里的不同应用，采取不同的存储策略，对于性能要求很高，或者是说很关键业务的应用，可能需要不同的存储资源。VeSpace Server SAN 可以根据不同业务的需求分配不同的存储资源，做到真正的应用感知。

场景 2: 虚拟桌面基础设施 VDI

虚拟桌面基础架构(VDI)，有更简单的系统管理、集中的安全性和数据保护。VDI 的存储可以使用块(存储区域网络(SAN))或者文件级协议(网络直连存储(NAS))。但是，虚拟化环境也给存储管理者带来了不小的挑战。包括基础设施成本增加、启动风暴等问题。



VeSpace Server SAN 提供自动精简配置，可以帮助 VDI 减少基础设施成本。VDI 可以通过本地访问和 IPSAN 标准协议访问 VeSpace Server SAN 提供的存储卷，VDI 管理平台可以自由的选择计算和存储的分离与融合两种不同的部署方式。

版权所有 ©深圳市云舒网络技术有限公司

场景 3: 超融合架构 HCI

超融合架构 HCI 指在通用的 x86 架构下，借助 SSD、10 GbE 网络等高速硬件，通过软件的方式实现存储池化、快照、克隆、分层、精简配置等企业级的数据功能。软件定义分布式存储是超融合的核心，VeSpace Server SAN 可以作为超融合架构 HCI 存储后端使用，为超融合架构 HCI 提供完整的分布式存储解决方案。

场景 4：数据保护端场景

用户使用存储时，在多用户登陆及复杂设备网络布局时，可能会面临权限隔离的问题，同时对数据的容灾和灾难恢复也有需求。

VeSpace Server SAN 支持多租户和保护域，提供权限和物理上的保护。可自定义多副本策略，数据分片会自动分布到不同服务器上；即便部分服务器损坏，也不影响数据可用。VeSpace Server SAN 提供了自动重建、快照、备份等多重措施，保护数据的有效性。

场景 5：数据库场景

数据库用户需要高性能和低延迟的块存储，同时对数据可靠性也有要求。而传统的 SAN+服务器的数据库应用存在集中式的机头 IO 瓶颈，扩展困难同时也无法提供更快的响应和更高的带宽，同时可靠性也主要依赖于单机存储的 RAID 技术，发生故障后重建较慢。

1. VeSpace Server SAN 能提供分布式的高性能和低延迟的块存储，不存在 IO 瓶颈，性能和容量可随存储主机增加而线性提升。
2. VeSpace Server SAN 存储采用分布式设计，请求可在多副本间智能并行处理，适用于 OLAP（高带宽）、OLTP（高 IOPS）的数据库应用。
3. 多副本和数据分片使数据实现高可用，即使发生故障，也可以多节点的快速重建。

场景 6：数据分析场景

大数据应用通常需要容量巨大、可扩充的存储空间，并且也需要较好的计算性能。

VeSpace Server SAN 部署在通用 x86 架构服务器上，支持存储计算合二为一，支持 HDD、SSD、PCI-E SSD 等存储介质。可线性扩展存储容量和计算性能，支持 Hadoop 等大数据应用。

版权所有 ©深圳市云舒网络技术有限公司

1.3 软件部署

VeSpace Server SAN 建议部署在>=3 台服务器环境中，下面描述具体模块的部署。

1.3.1 驱动层/控制层部署

驱动层(Driver)和控制层(Controller)必须融合部署，即部署在同一台主机上。Driver 和 Controller 在部署数量上不受限制，可以根据业务要求进行部署。

其中驱动层向上可以无缝对接 Docker、Swarm、KVM/QEMU 和 OpenStack Cinder，如果在应用场景中需要对接上述应用，则需要在 Driver 所在主机上部署上述应用。

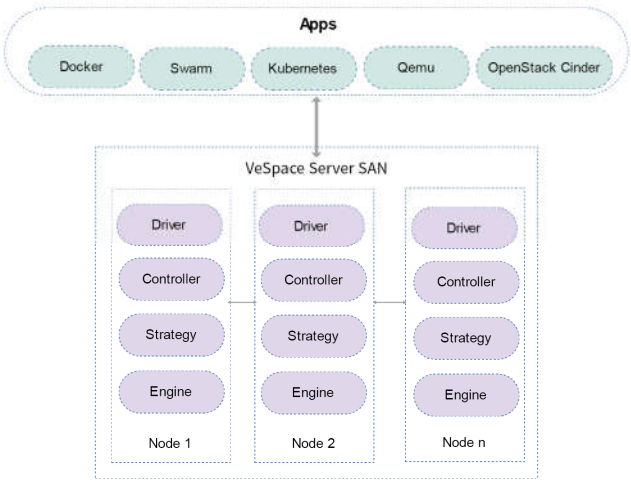
Driver 和 Controller 服务实际部署过程非常简单，只需要在符合要求的硬件平台下启动 VeSpace Server SAN 的 Driver 和 Controller 即可。其中硬件平台具体要求请参见 6.1 硬件平台章节。

1.3.2 策略层部署

策略层（Strategy）是 VeSpace Server SAN 存储系统的核心模块，负责整个存储系统的元数据管理、资源调度、策略控制等功能。策略层服务可以和驱动层、控制层、引擎层融合或分离部署。策略层服务推荐部署在 3 台服务器环境中，具体部署过程十分简单，启动 VeSpace Server SAN 系统策略服务即可。

1.3.3 引擎层部署

引擎层负责接收控制层发来的具体 IO 读写请求，各种存储数据的落地和存储特性的具体实现。引擎服务可以和驱动层、控制层、策略层融合或分离部署。原则上引擎服务在部署节点数量上没有限制，可以根据具体业务要求进行部署。部署过程十分简单，启动 VeSpace Server SAN 引擎服务即可。



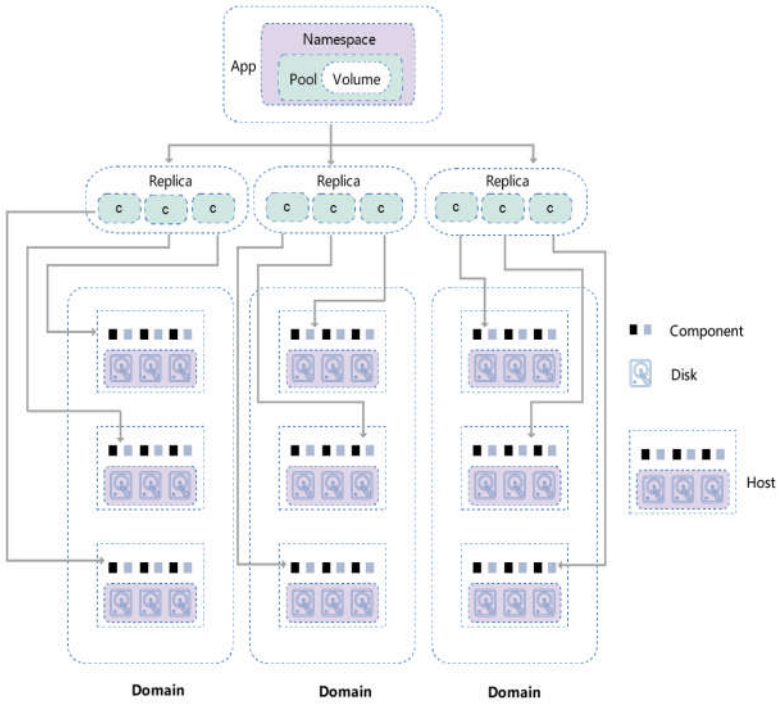
融合部署示意图

版权所有 ©深圳市云舒网络技术有限公司

二、VeSpace 策略子系统

2.1 元数据

2.1.1 元数据组织示意



为了实现在策略子系统中对整个 VeSpace Server SAN 系统进行资源调度策略控制、状态监控等核心功能，从物理角度和逻辑角度设计和组织了如上图所示的元数据层次结构。

从物理角度来看，为了使不同的存储主机和存储资源散落到不同物理区域，不同的物理区域相互隔离，在出现意外或灾害时能够有效的保证存储数据是可用的，策略子系统定义了保护域 Domain 这个元数据类型，在 Domain 下可包含多个存储主机 Host，在存储主机下又包含多个存储逻辑资源 component，这样就从物理角度把整个存储系统通过 Domain、Host、component 的元数据结构有效的组织了起来，方便了策略子系统对存储系统的管理和调度。

从逻辑角度来看，为了使得各个用户对存储资源的使用相互隔离、互不影响，策略子系统定义了命名空间这个元数据类型，在命名空间下可包含多个存储资源池 Pool，在存储资源池 Pool 下又包含多个存储卷 Volume，而为了保证存储卷 Volume 的数据高可用，在 Volume 下又设计了数据副本 Replica 这个元数据类型，每个 Volume 都可包含一个或多个 Replica，最后每个 Replica 下就对应一个或多个存储资源 component。至此，整个系统

版权所有 ©深圳市云舒网络技术有限公司

的元数据组织结构就如上图所示，正是有了这个树形的元数据组织结构，才能在策略子系统方便、灵活、高效地管理和调度整个存储系统的资源。下面针对每个元数据类型做简要说明。

元数据型	类型描述
Domain	保护域，物理隔离存储主机和存储资源。
Host	存储主机，每个存储主机必须划分到一个保护域，里面包含该主机所有的存储资源。
component	存储资源，最小的存储单元（逻辑上）。
Namespace	命名空间，位于元数据组织结构的最顶层，可包含多个存储资源池，用于各用户之间存储空间的相互隔离。
Pool	存储资源池，存储资源池可使用自定义的存储策略，资源池中包含多个存储卷。
Volume	存储卷，归属于存储资源池，可拥有多个存储数据副本，可保证数据的高可用。
Replica	存储卷数据副本，代表 Volume 的一个副本。

2.1.2 基于 Raft 算法实现的分布式 KV 模块 RStore

从上述整个系统设计的元数据组织结构来看，使用 Key/Value 键值对的方式进行元数据的存储是非常高效的，同时为了保证元数据的一致性和可靠性，我们开发了一个基于 Raft 算法的分布式 KV 模块 RStore 来管理和存储我们整个系统的元数据。同时针对我们系统具体的使用场景对 Raft 算法处理流程进行了一定的适配优化。

2.1.2.1 元数据一致性

分布式 KV 模块 RStore 是 Raft 算法在策略子系统中的实现和应用，并且保障在实际分布式存储场景中数据的一致性和异常数据恢复。RStore 模块目前设计实现的基本目标有以下几点：

（1） 具有容错性，保证异常情况下系统元数据的一致性
（2） 支持自动增删节点
（3） 支持增删查改操作

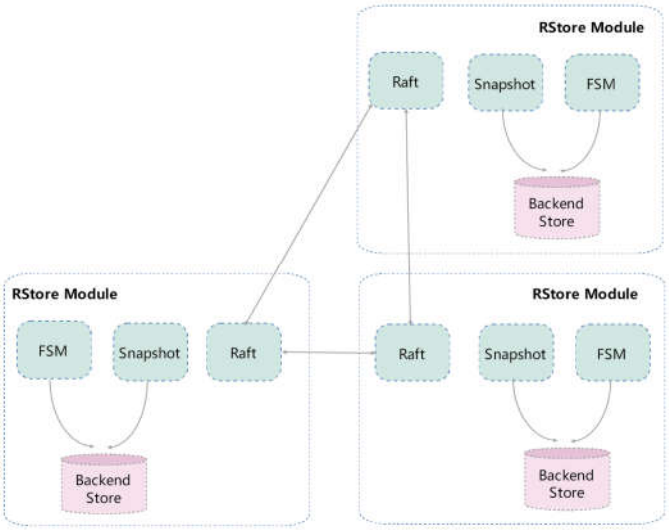
版权所有 ©深圳市云舒网络技术有限公司

（4） 支持适配多种后端存储引擎

（5） 支持日志快照重建功能

（6） 支持节点异常宕机系统元数据自动恢复

在 RStore 模块中我们主要实现了 Raft 一致性模块、日志状态机模块（FSM）、快照对象模块（Snapshot）和后端存储模块（Backend Store）。如下图所示：



从上图中可以看出，元数据分布式 KV 存储模块 RStore 主要包括 FSM、Snapshot、Raft 和 Backend Store 四个主要部分，其中 FSM 是 RStore 模块中日志的状态机，Snapshot 是元数据的快照对象，Raft 就是 RStore 模块中 Raft 一致性算法的实现（策略子系统元数据的一致性主要由 Raft 模块来保证），Backend Store 为元数据的索引和存储模块（负责元数据从内存到磁盘的实际落地）。下面就 RStore 模块如何保证元数据一致性、故障恢复、快照和存储落地做简要的说明。

首先 Raft 一致性算法在确保分布式存储系统数据一致性方面已经非常成熟和可靠，也在业界得到了广泛的应用和验证，因此 RStore 模块选择实现 Raft 算法来确保策略子系统元数据多副本的一致性。在系统正常运行过程中，RStore 中的 Raft 模块会进行如下处理流程来保证元数据的一致性。

（1） 在某一时刻策略系统集群子节点成为一个候选者(Candidate)，它会在一个 TERM 内向其他 Follower 发出选举投票请求，请求 Follower 选择自己成为 Leader。

（2） 其他 Follower 的 Raft 模块会向发起投票的节点进行回应，若同意则回应 YES，不同意则回应 NO，在整个投票过程中，由于各个节点可能会故障或者宕机，也有可能有其他的 Candidate 发起投票，但只要 Candidate 的最后选举票数达到 N/2 + 1（N 为策略子系统的节点数），那么该 Candidate 就可以成为 Leader。

版权所有 ©深圳市云舒网络技术有限公司

（3） 选举投票结束后，成为 Leader 的节点就可以向其他节点发出日志复制的等操作并会与其他节点之间维持一个心跳来感知节点之间的状态。这时对于系统元数据的修改都先经过 Leader 节点，每个节点都会写一条日志记录，Leader 会复制该条日志记录到所有 Follower 上，等到大部分 Follower 都响应时才提交日志，并通知 Follower 日志已提交，所有的 Follower 收到通知之后也在各自节点上提交该条日志，然后每个节点根据日志记录将元数据的修改实际存储落地。这样就保证了系统元数据是一致的。

（4） 若在系统运行过程中，Leader 节点故障或者宕机，导致整个系统没有存活着的 Leader 节点，那么系统以后的修改操作就将没有 Leader 节点来带领其他节点进行元数据修改操作。发生这种情况后，RStore 中的 Raft 模块由于在一定时间内没有收到 Leader 节点的心跳，此时其他节点就会认为 Leader 已经失效，会有新的节点成为 Candidate 发起新一轮投票，具体的投票处理流程和前面的投票流程完全一样，直至选出新的 Leader 继续带领其他节点进行系统元数据的修改操作。因此，系统在运行过程中，规定数量范围内的节点宕机，元数据一样可以保证一致并不会影响整个策略子系统的正常处理。

（5） 脑裂情况。选举过程是有一定的时间限制，如果恰巧有两个节点同时成为 Candidate，同时向其他节点中的 Raft 模块发起投票请求，这时就会出现脑裂投票的情况，两个节点都可能拉到一样多的选票，本轮选举失败，没有选举出 Leader，那么此时 Raft 模块中会内置一个计时器，最先超时的 Follower 会马上再次成为 Candidate，并发起新一轮的投票，由于 Raft 模块中超时时间是一个随机数，同时超时的概率会非常小，因此系统总能保证在多轮投票之后选举出新的 Leader，从而带领其他节点进行元数据修改操作，保证系统元数据的一致性。

（6） 若发生网络分区，在网络分区消失后，系统同样可以达成整个系统数据是一致的。假设策略子系统有 A-E 五个节点，B 是 Leader，如果发生网络分区，A，B 成为一个网络子分区，C、D、E 成为一个网络子分区，此时 C、D、E 节点 RStore 中的 Raft 模块之间会发起新的选举，假设选举出 C 作为新的 TERM 的 Leader，这样在两个网络子分区内就有了不同 TERM 的两个 Leader，这是如果有客户端写入 A 时，因为 B 无法复制到大部分 Follower 中，所以日志一直会处于未提交的状态，同时另一客户端对 C 节点的写操作能够正常完成，因为 C 是新的 Leader，此时知道 D、E 节点。那么当网络恢复时，B 节点能够发送心跳的给 C、D 和 E 了，发现 C 节点的 TERM 值更大，这时 B 节点就会自动降级为 Follower，然后 A 和 B 节点都会回滚未提交的日志，并从新的 Leader 也就是 C 节点那里复制最新的日志记录并进行相应操作，因此通过各个节点 RStore 中 Raft 模块之间的交互保证了在出现网络分区时系统元数据的一致性。

版权所有 ©深圳市云舒网络技术有限公司

2.1.2.2 元数据存储

在 RStore 中的 Snapshot 模块是为了防止系统的修改数据过多而进行的状态快照，状态是由 RStore 中的 FSM 状态机触发的，FSM 根据设置在系统的修改数据状态达到一定的阈值之后触发状态快照。做了状态快照之后，就会将快照之前的状态数据进行删除，节省存储空间，同时所做的数据快照也可用于状态恢复和回滚。当服务异常重启后，RStore 会检测是否存在快照，如果存在快照，则快照的数据会进行重建，RStore 的 Raft 模块就会根据日志的索引重放那些没有在快照之内的日志条目，这样系统的元数据就快速地得到了恢复。

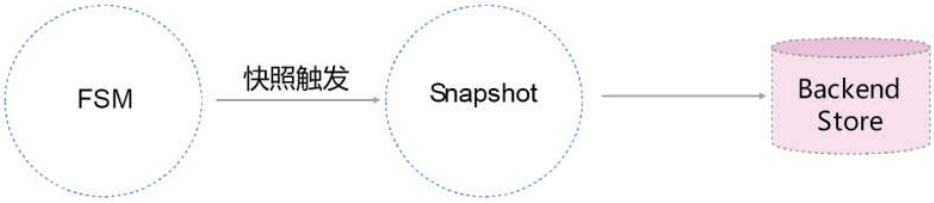


图 元数据快照示意图

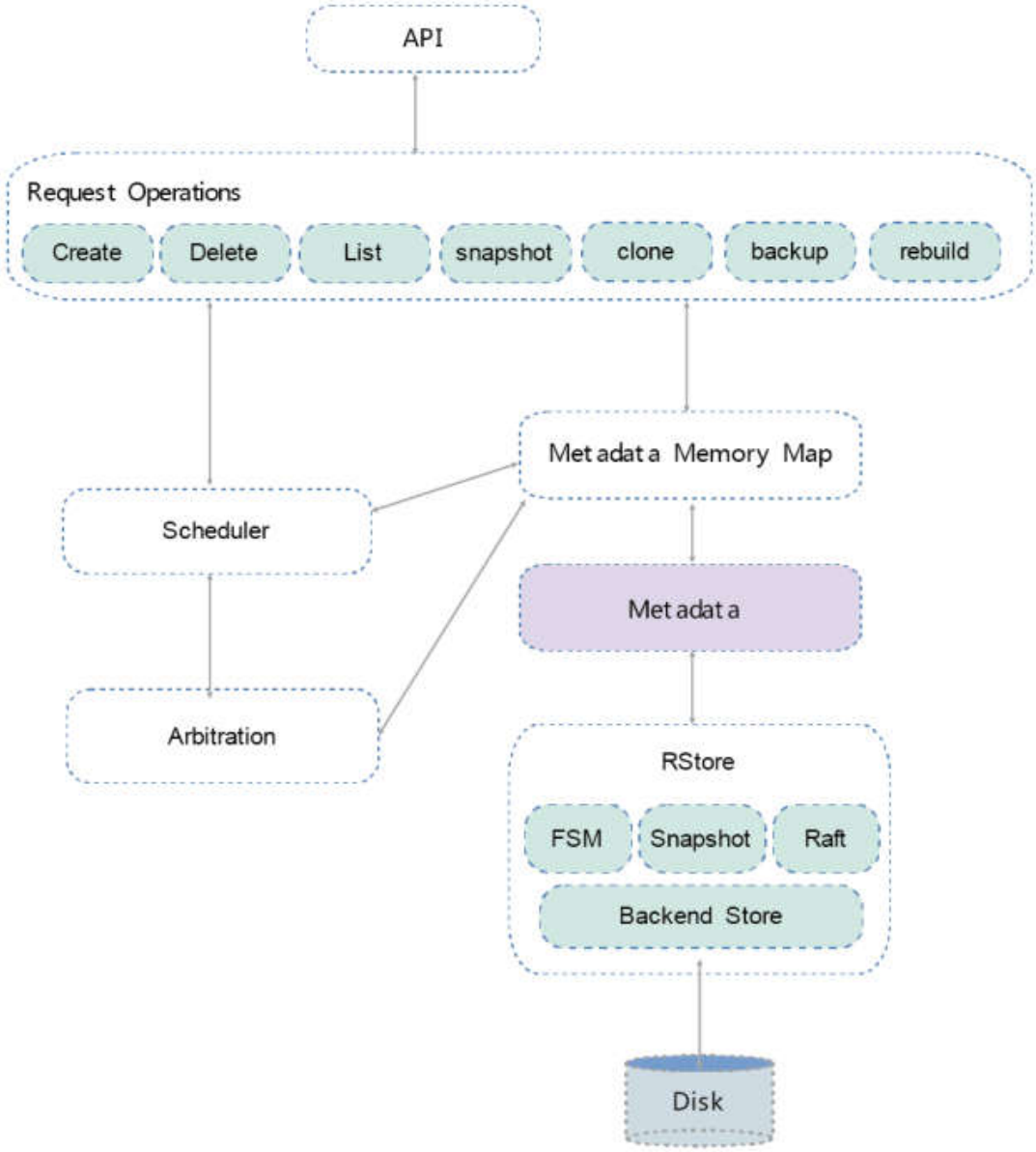
Backend Store 模块是 RStore 中元数据索引与存储模块，它负责元数据的内存索引和数据落地。在 RStore 中，采用元数据逻辑存储和实际底层存储分离的机制，Backend Store 模块既可以处理所有的存储逻辑，同时也实现了一个新的存储后端，可以支持二进制和多版本的键值对存储，并支持具有低内存使用率的增量快照，最后负责元数据的持久化（即磁盘上）存储。在实现过程中采用分层排序的 BTree 数据结构来作为元数据存储的内存索引，并实现了 Bitcask 键值存储模型来作为元数据持久存储层。其中 BTree 中的每个节点就代表着一个元数据，节点值就指向该元数据在磁盘上的实际值。对于每个元数据都会保留其值的所有先前数据版本，因此可以很方便地查看系统中元数据的修改记录。由于在实际应用场景中，对系统元数据的读操作会非常频繁，因此会在系统中构建元数据的内存映射 ClusterPhysicalMap 和 ClusterLogicMap。以此来加快系统对元数据的访问。具体见下节。

最后，策略子系统中的基于 Raft 算法的分布式 KV 模块 RStore 对外提供标准的 KV 接口（Put、Get、Delete 等），同时也提供元数据的事务性操作接口，可接受标准的 RESTful API 请求来访问系统的元数据，也提供元数据批量操作的接口。因此，RStore 对于元数据的访问是非常方便，简单灵活和可靠的。

版权所有 ©深圳市云舒网络技术有限公司

2.1.3 元数据在策略子系统中位置

在策略子系统中元数据模块是整个系统资源调度、策略控制的基础，正是因为有了元数据，系统才能实现对整个系统的有效管理。元数据模块在整个策略子系统中的位置如下图所示：



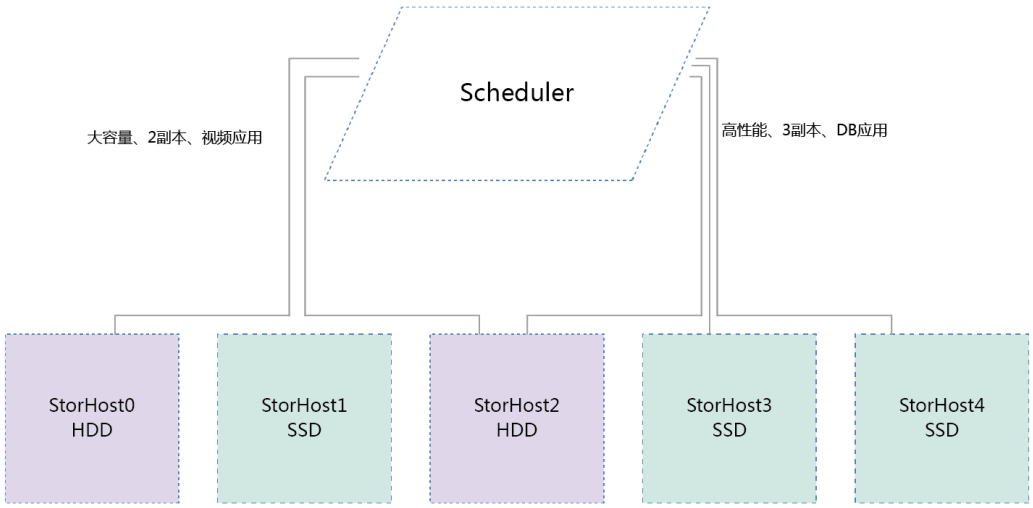
2.2 存储策略

策略可以使用户根据自己的需要或业务偏好，设置若干过滤项及权重值，由调度器选出最合适的存储主机。

策略子系统提供了副本数、过滤、权重三类可选策略，用户在创建策略时可自由选择其中任意项。过滤参数是可选的，用户可根据业务需要，例如调度出高性能、低负载或大内存

版权所有 ©深圳市云舒网络技术有限公司

的存储服务器来进行设置，不设置则不生效；可选权重参数和默认权重一起在调度时对可选主机进行排序，选出最优存储主机。



图例是两种场景策略下的不同调度结果。

策略需在创建存储池前添加，在创建存储池时指定某一策略，策略会作用于该存储池下所有卷。策略决定调度器的行为，例如副本数可以让存储池下的卷都拥有指定个数的副本，而不同的过滤及权重也会使调度器返回不同的调度结果，细节请看调度。

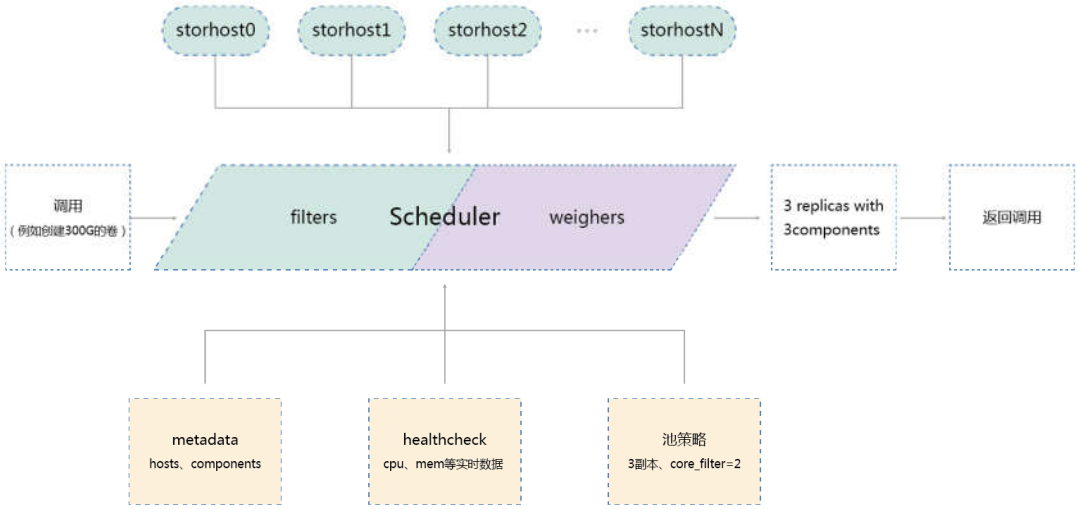
2.3 调度

调度器（scheduler）的功能是为各基础操作（新建、扩容、克隆等）调度出可用的后端存储主机。调度器根据用户设定的卷策略，再加上实时的监控信息，从所有可用的存储主机中调度出最合适的 component 以供使用。

调度主要由两个部件组成，一个是过滤器（Filter），一个是权重器（Weighter）。两者的输入都是可用的存储主机列表；首先根据策略指定的过滤参数，filter 先运行筛选出合适的主机；然后根据策略指定的权重值，Weighter 进行排序，最终选出最合适的存储主机。

从整体来看调度过程如下图：

版权所有 ©深圳市云舒网络技术有限公司



从局部来看，单个 component 的调度过程如下图：

最后调度是以 component 为单位的 ,即用户创建一个 3 副本 5component 的存储卷，scheduler 会内部调度后一次性返回结果给调用程序。这样的设计能保证：

1. 对创建卷的全部调度要么是成功的，要么是失败的
2. 组件分散较均匀，不会出现热点
3. 至少有一个保护域拥有独立、完整的副本

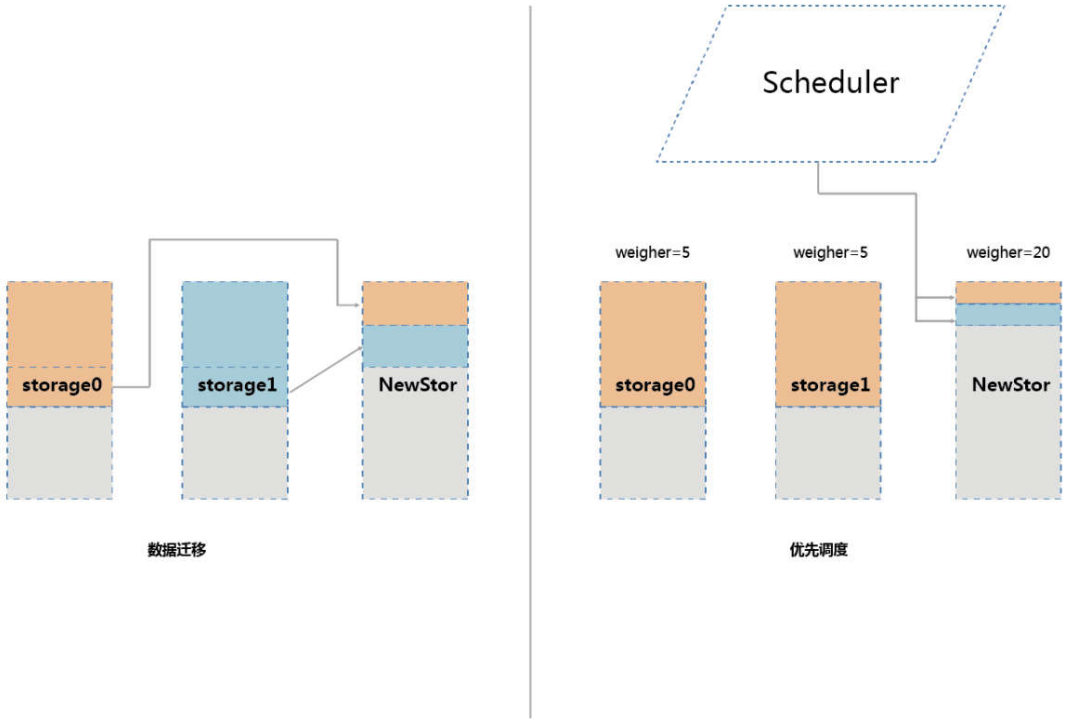
版权所有 ©深圳市云舒网络技术有限公司

4. 相同 index 的 component 不会出现在同一台主机上，即使某台存储主机宕掉也不会导致多个副本失效到无法恢复

2.4 弹性扩展调度

2.4.1 数据分布

业界常用的分布式算法，不论是 CRUSH 还是一致性哈希，在新存储加入时都伴有移动已有数据来达到平衡的做法。例如已有 2 台 100G 的存储，达到 60%写入量后新增了 1 台 100G 的存储主机，整个集群可能会有 40G 的数据发生迁移。然而实际应用中存储容量几十上百 T 也是很常见了，这样会占用了大量的后台网络带宽和计算资源。策略子系统作了免迁移的设计，添加了新的存储后，新创建的卷在经过调度后会尽可能地落在新加入的存储主机上，从而无需数据迁移就能达到存储资源的有效利用。



2.4.2 平滑扩容

策略子系统具有良好的扩展性，支持动态的扩展存储容量，且无需中断应用的运行。系统可以随时添加新的存储主机，不论是 VeSpace Server SAN 还是 CEPH、NFS 等常见存储资源，新添加的存储均可以立即被调度系统应用到接下来的调度中，新的创建等请求会出于一定的优先原则落在新添加的存储上，平滑的生效于接下来的应用中。因此，用户后期可以灵活的根据业务需求添加新的存储服务器以扩容存储空间和增加计算能力，而且随着存储

版权所有 ©深圳市云舒网络技术有限公司

服务器的增多，整套系统的聚合带宽也会线性、无缝的增加，完全可以满足业务不断发展所产生的容量和性能需求。

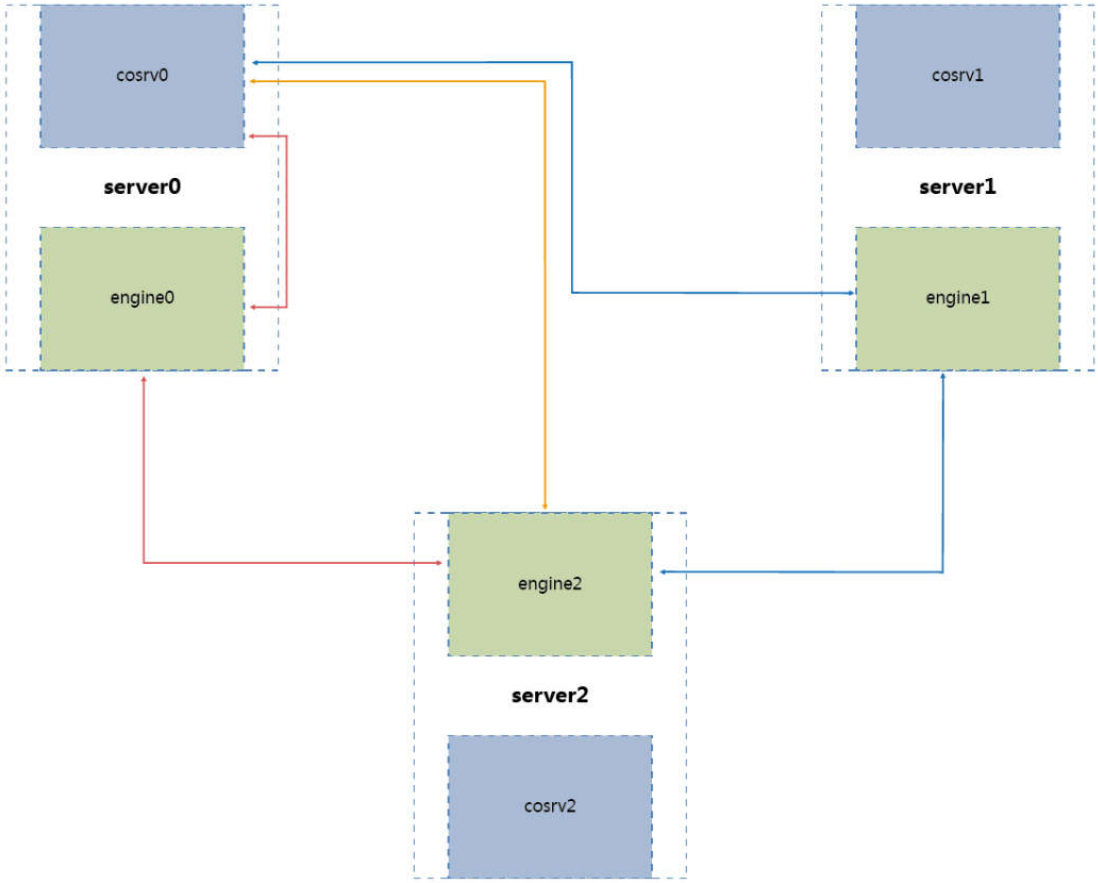
2.5 仲裁机制

在集群中主机间想要知道对方的状态，同时要防止脑裂情况下的误判，这时候就必须要依赖仲裁机制来对状态做出一个正确的判断。仲裁实际上是一个多数对少数的投票，关键在于选取哪些主机作为 witness。

仲裁采用了类似 vSAN witness 仲裁机制的方法，利用 VeSpaceServerSAN 集群的主机来充当 witness，根据多数有效投票来仲裁目标的状态，可有效避免脑裂和孤岛。

当策略主机需要知道某一节点的状态，会将仲裁请求发往集群中的所有 n 个同类节点，汇总投票结果来得出仲裁结论。当 n 为偶数时，还会将仲裁请求发往其他策略主机（不包含自身，故为 n-1 个），再次汇总所有投票结果，从而得到有效投票，得出仲裁结论。

以一个典型 3 节点的集群来举例：



版权所有 ©深圳市云舒网络技术有限公司

策略层节点 0(cosrv0)想要知道存储节点(engine2)的状态，它会将仲裁请求发往集群中的全部 3 个 engine 存储节点，然后回收投票结果，进而仲裁出 engine2 的状态。

2.6 高级特性

2.6.1 克隆

VeSpace Server SAN 提供了克隆功能，支持基于一个快照创建出来多个克隆卷，各个克隆卷刚创建出来时，数据内容和卷快照中的数据内容是一致的，后续对克隆卷的修改不会影响原始卷的快照和其他的克隆卷。

所有克隆卷继承了原始卷的所有功能：克隆卷也可以像原始卷一样再次被快照和克隆。

2.6.2 重建

VeSpace Server SAN 将集群中的各个节点上的存储资源抽象成一组 Component，这些 Component 的副本被相应的策略分布在不同的存储节点上，当存储节点中发生硬盘(SSD、HDD)或者节点故障的时候，策略子系统会自动发起重建，保证丢失数据在短时间内恢复，由于数据存储单位是 component，因此重建的时候，只需要故障相关的若干 component 即可，不需要校验整个卷的数据。这样重建过程非常快速。

根据故障类型，我们把重建分为两种类型:

- 局部重建：当存储主机，网络发生异常，计划性重启等情况，引起 component 短时间内失效，导致卷的副本间少量数据不一致时，策略子系统重建控制模块，检测到异常情况，进入局部重建阶段，将少量不一致数据进行修复，保证副本数据的一致性。
- 完全重建：当存储主机永久性宕机，磁盘替换，网络长时间异常等情况下，component 长时间失效，策略子系统重建控制模块为了避免副本损坏造成数据安全性降低的风险，进入完全重建阶段，将在集群的其他存储主机上，重建失效的 component，确保全部副本可用。

重建的触发是由策略层来主导，具体操作需要后端和前端配合做处理。

2.6.3 自动迁移

版权所有 ©深圳市云舒网络技术有限公司

VeSpace Server SAN 的数据卷默认是自动精简配置的，单个磁盘可以置备比其空间大50%的总空间，当单个磁盘的使用量超过其预设阀值时，策略子系统自动迁移模块将智能挑选该磁盘内合适的 component 迁移到空间足够的磁盘中，达到集群数据总体的均衡。

当某个磁盘成为访问热点时，策略子系统自动迁移模块将挑选合适的时间点(避开高峰期，自定义)，触发自动迁移流程，达到集群访问负载均衡。

2.6.4 扩容

当卷的存储容量不够时，可以通过扩容功能来扩充原线性卷的容量。扩容发起后，策略层会更根据卷的属性及策略调度出新的 component 附加在各个副本的末尾，这样该卷能够平滑地扩容至新的大小，且不影响已有的数据。

2.6.5 备份

VeSpace Server SAN 除了在本地提供健壮、可靠的数据备份外，还可以备份数据到第三方的公有云上，以及从公有云上的备份恢复出本地数据。同时备份策略是异步的将已写入的数据备份至云端，而不会因同步未写入的部分而耗费时间空间。

2.6.6 只读

VeSpace Server SAN 的卷除了回滚数据到指定快照点外，还有一种简便有效的方法查看历史快照版本的数据。通过只读模式切换到某快照点，可以快速方便的查阅历史数据，而不用担心做快照影响当前业务及污染历史数据。

2.6.7 校验

VeSpace Server SAN 提供自动及手动触发 scrub 的两种方式，校验多个副本间已写入的数据是否一致。如果出现不一致的情况，会自动将失效 component 加入重建并通知用户。

2.6.8 多路径

VeSpace Server SAN 的 SCSI 块设备和 iSCSI Target 设备支持多路径功能，通过将卷挂载到多台控制主机，用户将能从多个节点访问到同一个卷的数据，提高可用性。

版权所有 ©深圳市云舒网络技术有限公司

2.6.9 切换设备类型

VeSpace Server SAN 支持用户切换现有模式的卷到新的设备访问类型，设备类型可在 SCSI 块设备、iSCSI Target、NFS 共享、SMB 共享、Rest 共享间自由切换，随时改变对存储卷的访问方式，支持同时提供多种共享方式访问。

2.6.10 IOPS QoS

VeSpace Server SAN 支持用户制定卷的 IOPS 峰值限制，可有效防止由于个别卷偶发的高负载 IO 峰值导致其余数据卷访问性能骤降的问题。并且用户可自行根据不同性能的存储池，制定不同的工作负载策略。

2.7 集群监控

通过在每台主机上部署监控服务进程来收集并按需存储整个集群各种服务的状态，主机 CPU、内存、负载、网络、磁盘 IO 等系统参数，同时可以实时监控存储集群的存储空间使用情况，如集群总容量、已用容量、可用容量、存储池、数据卷等统计信息，并可根据监控信息推送告警。监控服务采用插件化设计方式，可以很方便地进行扩展，监控节点可扩展至任意多个，并能将监控数据持久化保存，用户可查看集群监控历史数据。

2.8 日志和告警管理

2.8.1 日志管理

对于集群里的所有主机都提供日志收集功能；日志主要包括 VeSpace 系统各个服务的运行日志，日志的内容主要记录服务在每个时间点的访问和操作结果，日志级别分为 error、debug、info 等；日志记录提供查询机制；日志记录空间满时提供自动删除和转存机制；日志时间提供统一的时间源机制；同时集群为用户提供一键收集一个或多个主机的日志并下载收集的日志，为用户提供管理和分析的依据。

2.8.2 告警管理

集群提供告警设置、告警推送等管理功能；用户可以设置告警的触发条件和告警推送的方式；对于系统出现的异常状态和各种故障，可实时显示在操作界面，同时用户进行告警恢复；支持 Email 告警、短信告警、Trap 告警。

版权所有 ©深圳市云舒网络技术有限公司

三、VeSpace 驱动子系统

3.1 驱动容器

目前越来越多的企业将服务进行容器化部署，这样可以极大地简化企业部署和运维成本，而 Docker 容器引擎则是目前最流行和使用度最高的容器引擎，因此驱动子系统支持无缝对接Docker容器引擎 ,使得各种容器化服务场景可以很方便高效地使用VeSpace Server SAN 存储系统。

尽管 Docker 容器的根镜像支持各种驱动，如 aufs、device mapper 等，但都不是针对存储卷 Volume 的，虽然用户可以通过 docker run –v 来创建卷，但是这种方式创建的卷是一次性的 ,并不能轻易地重用于其他主机上的新容器。例如 ,在主机上启动一个 MySQL 数据库服务容器，写入一些数据，删除该容器之后，修改的数据库数据将不能跨主机迁移和重用。

驱动子系统使用 Docker 卷插件机制为 Docker 容器提供持久性存储卷，并支持各种存储后端和卷创建、删除、挂载、快照、备份和恢复等功能。使用 VeSpace Server SAN 用户能够在主机之间任意迁移存储卷，跨主机共享一个或多个存储卷，根据需求制定存储卷的快照以及回滚到特定版本的快照。这对于用户来说，使用驱动子系统将会使自己的容器化服务对后端存储卷的数据访问和管理变得更加容易和灵活。除此之外，驱动子系统还支持无缝对接 Docker Swarm 和 Kubernetes。

3.2 驱动虚拟机

随着虚拟化技术的日益成熟，虚拟机的使用已经变得非常常见和流行，因此针对虚拟机的应用场景，驱动子系统可以无缝对接各种虚拟机管理系统，如 KVM/QEMU 等；为虚拟机提供可靠的存储资源。用户在部署 VeSpace Server SAN 之后，就可以使用 VeSpace Server SAN 创建自己的虚拟机服务，并在使用过程中能够非常方便高效的将数据写入 VeSpace Server SAN 的存储卷中。

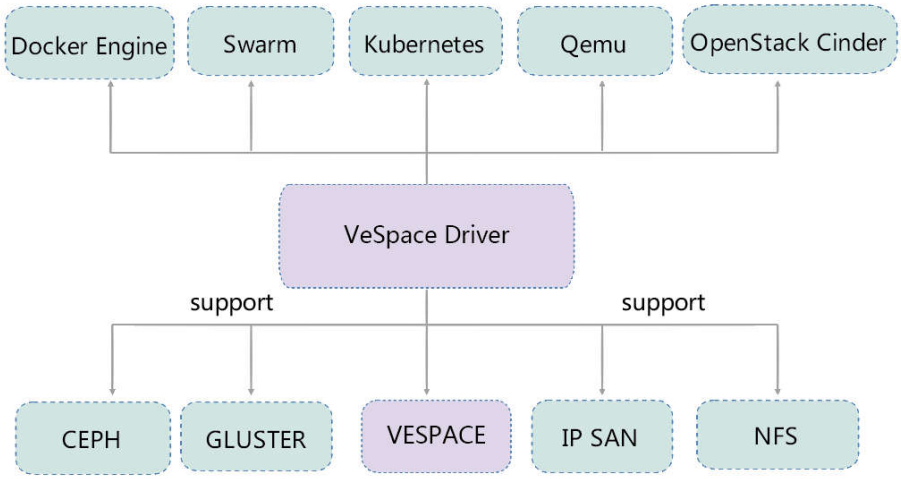
3.3 驱动 OpenStack

所谓驱动 OpenStack，其实对于存储来说，就是对接 OpenStack 云服务平台中的块存储服务组件 Cinder。目前几乎所有主流存储厂商都支持 OpenStack Cinder 服务，因为对接 Cinder 服务，就可以给虚拟机提供块存储服务，比如创建 Volume、挂载 Volume、创

版权所有 ©深圳市云舒网络技术有限公司

建快照等。因此 VeSpace Server SAN 系统也针对 OpenStack 应用场景，按照 OpenStack 社区规范实现了自己驱动 Cinder 服务的 Driver 和 Feature，并构建了相应的自动化测试。

3.4 驱动子系统框架图

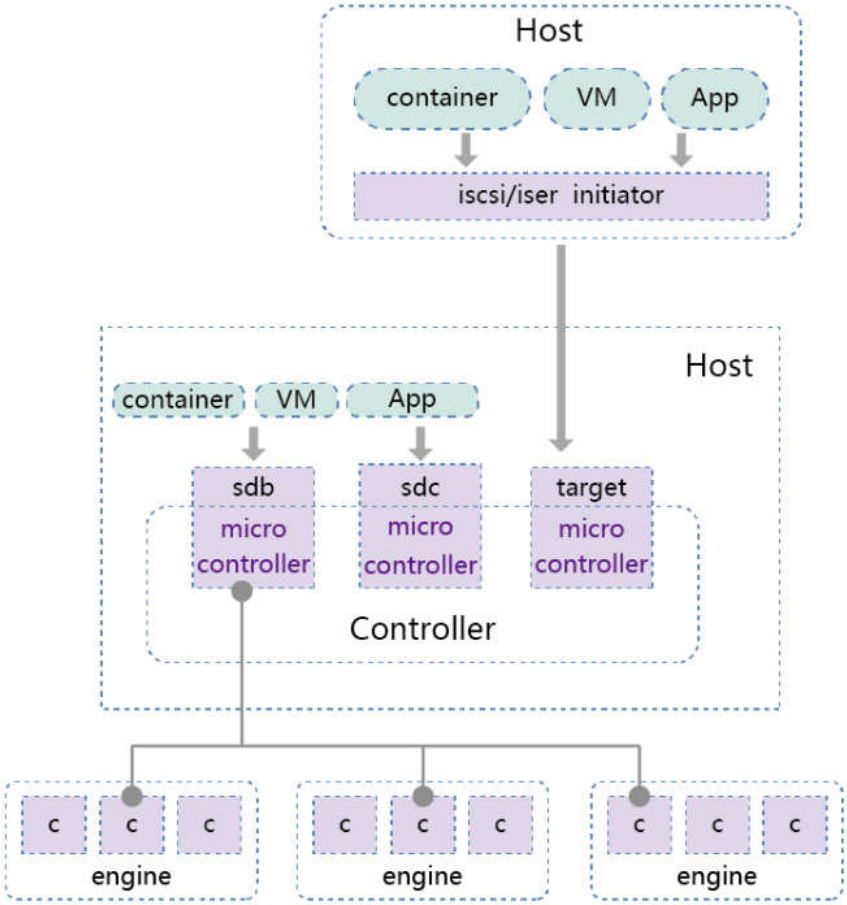


四、VeSpace 存储子系统

4.1 块接口

VeSpace Server SAN 以两种方式提供块接口：1.标准块设备；2.iSCSI/iSER target。如下图：微控制器可以提供标准块设备接口，容器、虚拟机、应用程序可以将卷当成本地磁盘使用；也可以提供 iSCSI/iSER target，容器、虚拟机、应用程序可以通过标准 iSCSI 和 iSER 访问虚拟块设备。

版权所有 ©深圳市云舒网络技术有限公司



4.1.1 标准块设备

支持标准 SCSI 协议，对于上层应用卷与本地磁盘在绝大多数场景下无任何区别，可直接对卷进行读写，亦可对卷格式化文件系统，挂载后，进行读写。

4.1.2 iSCSI/iSER Target

支持标准 iSCSI 协议，可通过 iSCSI initiator 在远端发现并使用微控制器生成的 iSCSI target。

支持 CHAP 身份验证以保证客户端的访问是可信与安全的。CHAP 全称是 PPP 询问握手认证协议（Challenge Handshake Authentication Protocol）。该协议可通过三次握手周期性的校验对端的身份，可在初始链路建立时以及链路建立之后重复进行。通过递增改变的标识符和可变的询问值，可防止来自端点的重放攻击，限制暴露于单个攻击的时间。

iSER(iSCSI Extensions for RDMA)是 iSCSI 使用 RDMA 的扩展协议，该协议允许网络数据直接与主机内存之间进行数据传输，而不需要数据拷贝和 CPU 介入。

版权所有 ©深圳市云舒网络技术有限公司

RDMA(Remote Direct Memory Access)远程直接数据存取，是应用程序内存与网络适配器直接传输数据的零拷贝(zero-copy)技术。

4.2 文件接口

VeSpace Server SAN 提供标准的 NFS、SMB 文件共享协议，满足标准 NAS 文件共享资源存储需求。

VeSpace Server SAN 提供了文件共享协议：

1.**NFS(Network File System)**：UNIX 系统使用最广泛的一种文件共享的客户/服务器协议。

2.**CIFS(Common Internet File System)** ：支持客户端通过 TCP/IP 协议对处于远程计算机上的文件和服务发起请求。

4.3 对象接口

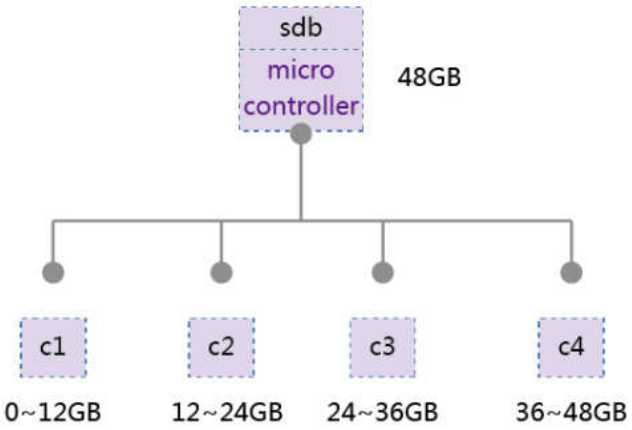
提供标准的 Restful API 通过 HTTP 协议访问存储资源 ,方便用户上传和下载文件对象。

4.4 卷类型

VeSpace Server SAN 支持三种不同类型数据分布的卷：1.线性分布；2.条带分布；3.EC 分布；

4.4.1 线性分布

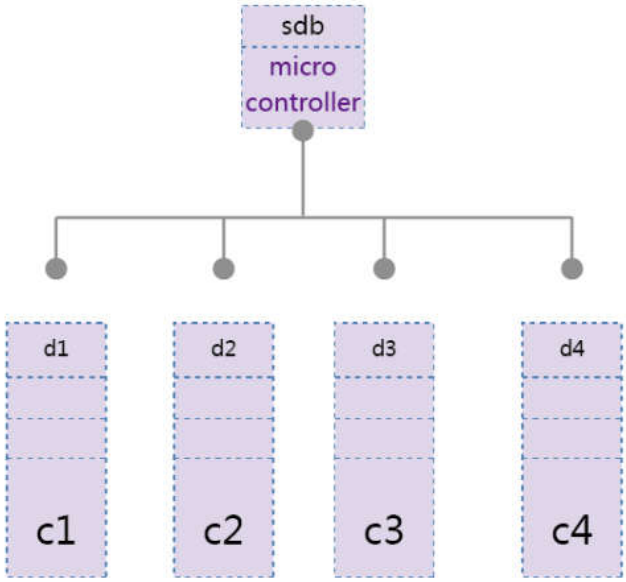
如下图，假设创建一个虚拟卷大小为 48GB，线性切分成 4 个 component，C1 负责线性地址空间 0~12GB-1，只要是对该地址范围内的访问，将请求交给 C1 处理。



版权所有 ©深圳市云舒网络技术有限公司

4.4.2 条带分布

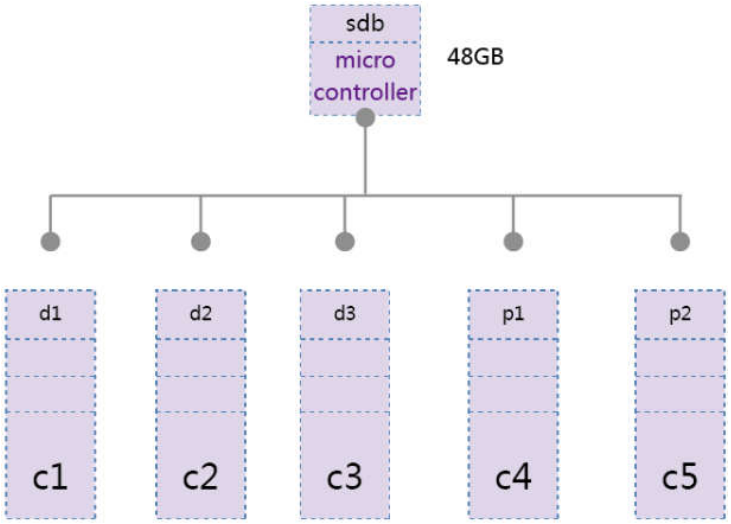
如下图，将数据切成大小相等的数据块(chunk)，所有 component 相同位置的数据块组成条带(stripe)，类似于 RAID0，跨数据块的读写，并行处理，提升 IO 速度，减少延迟。



4.4.3 EC 分布

纠删码(erasure coding，EC)是一种数据保护的方法，将数据切割成大小相同的数据块(chunk)，把通过计算得到与数据块(chunk)大小相同的校验块(下图中的 P1)。EC 通过时间换取空间的方式，通过增加 CPU 计算量，而减少数据的冗余。

如下图，卷 sdb(48GB)使用了 3+2 的纠删码，允许两个 component 失效，总使用空间为 80GB。如果使用副本的形式则需要占用 144GB 的存储空间，达到允许两个 component 失效的目标，纠删码节省了 80%的存储空间。

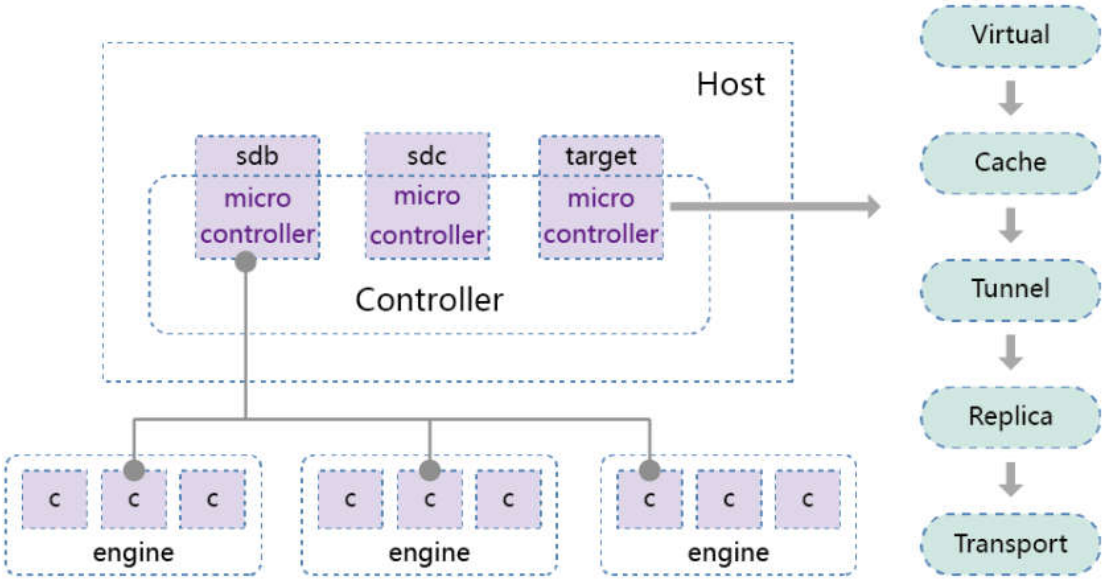


版权所有 ©深圳市云舒网络技术有限公司

4.5 微控制器

每个卷都有一个单独的微控制器(Micro Controller)，微控制器控制卷的一切，微控制器为独立进程，所以卷之间互相独立，互不影响。如下图，每个微控制器划分为 5 个层次：

- **Virtual**：虚拟层，创建，删除，管理卷，接收应用程序对虚拟块设备的读写请求，将请求转化后，提交给下一个层次。
- **Cache**：缓存层，在卷创建时指定(可以选择不使用缓存)，用于加速请求处理。具体的查看下面章节--应用端 Cache。
- **Tunnel**: 通道层，所有的 IO 请求必须通过 Tunnel，Tunnel 主要负责请求的转发和响应接收与处理，通过计算将请求发送给对应的 Replica，Tunnel 还负责请求超时控制，流控 QoS、异常处理、数据安全等级控制。
- **Replica**：副本层，Replica 的数据分布在多个 component，副本层中实现卷的数据分布算法，管理卷的多个副本状态，如果有异常发生(网络异常、后端主机异常等)需要让 Tunnel 及时知道，并进行异常处理。
- **Transport**：传输层，管理与 component 之间的 TCP 链接，进行链接时的验证与数据传输，实施监控 TCP 链接状态，如有变化，上报给副本层。



4.5.1 应用端 Cache

应用端 Cache 存在于虚拟块设备指定的空间中，如下图，cache 对于读写采用不同的机制。

版权所有 ©深圳市云舒网络技术有限公司

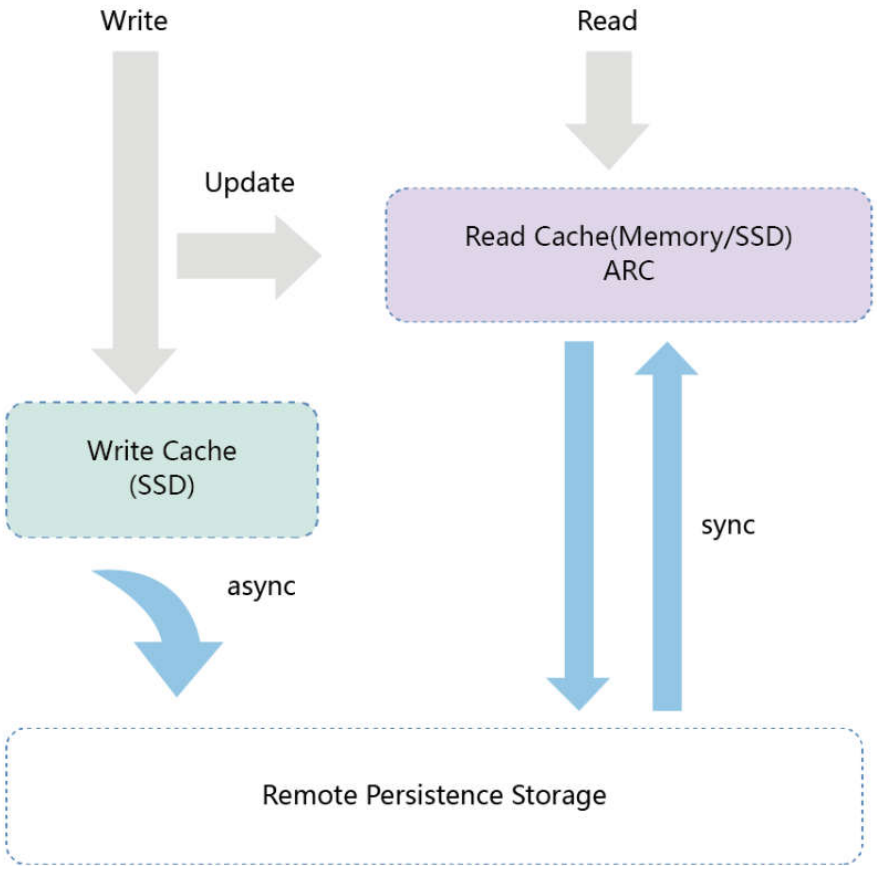
4.5.1.1 Read Cache 机制

读 cache 的命中率是提升读性能的关键点，这里采用了 ZFS 中读缓存算法 ARC(Adjustable Replacement Cache)。读 cache 介质可以是内存或 SSD。

当虚拟层 Virtual 向 Cache 层提交读 IO 操作，Cache 层根据 IO 的偏移量和大小查找是否命中读 Cache，如果命中，直接从 Cache 中将数据返回给虚拟层，完成此次 IO 操作。如果没有命中，将此次 IO 操作同步传递到后端将数据读取出来，返回给虚拟层，并更新到缓存，由缓存层决定是否将此次 IO 数据保存到读 cache 中。

4.5.1.2 Write Cache 机制

为了数据安全，写 cache 的存储介质必须是 SSD，并且需要指定允许脏数据的最大值，超过一定量的脏数据会触发写 cache 中的数据向后端 flush。在数据写入时，需要实时检查读 cache 中是否有该数据，如果有，则将其更新，避免读 cache 的脏数据导致应用程序的读取的数据不一致。



版权所有 ©深圳市云舒网络技术有限公司

4.5.2 QoS（IOPS 限制值）

QoS 作为存储系统的高级功能，市面上许多存在了几十年的存储阵列，支持 QoS 的也不多见，目前 VeSpace Server SAN 的 QoS 有如下特点：

- 支持以卷为单位的 QoS 限制值，支持以卷粒度提供满足性能的服务等级协议(SLA)。
- 卷使用的过程中修改 QoS 限制值，对应用程序无感知，使得主机 IOPS 压力过大时，可以进一步限制单个卷的 IO 上限。
- QoS 通过定时采集 IO 负载，根据自研算法动态计算 IO 的发送速率，从 IO 源头层面整体控制读写 IO 的行为，从而保证 IO 的性能一致性，防止 IO 出现不确定的抖动。

4.5.3 压缩传输

压缩传输指的是数据通过网络传输之前，将数据进行压缩，在 engine 收到数据时，对数据进行解压，以降低网络开销，对于网络环境较弱，或异地之间传输大量数据时，压缩传输非常凑效。

VeSpace Server SAN 以卷为单位指定压缩传输 ,在创建卷时 ,指定是否需要压缩传输。压缩传输在弱网络环境下，能获得 30%左右的性能提升。

4.5.4 数据安全等级

在 CAP 理论中，一致性，可用性和分区容忍性三者不能同时兼得，但在真实使用时，对于不同的应用程序，有些能够舍弃一致性获得可用性，有些却要保证强一致性，面对这些需求，VeSpace Server SAN 将这个选择权交还给用户。

VeSpace Server SAN 支持以卷为粒度的数据安全等级设置，目前将安全等级划分为 3 等，由用户在创建卷时指定：

- 等级 1（一般）：只要有一个副本返回成功，表示本次写 IO 操作成功，如果有失败的情况，由分布式系统自动修复，弱一致性，但只要有一个副本可用，则卷可用，确保业务连续性。
- 等级 2（高）：有一半以上的副本返回成功，表示本次写 IO 操作成功，如果有失败的情况，由分布式系统自动修复，达到最终一致性，并且满足 Quorum 机制；一半以上副本可用，卷可用。

版权所有 ©深圳市云舒网络技术有限公司

- 等级 3（最高）：必须所有副本返回成功，表示本次写 IO 操作成功，只要有一个副本写失败，此时卷不可用。

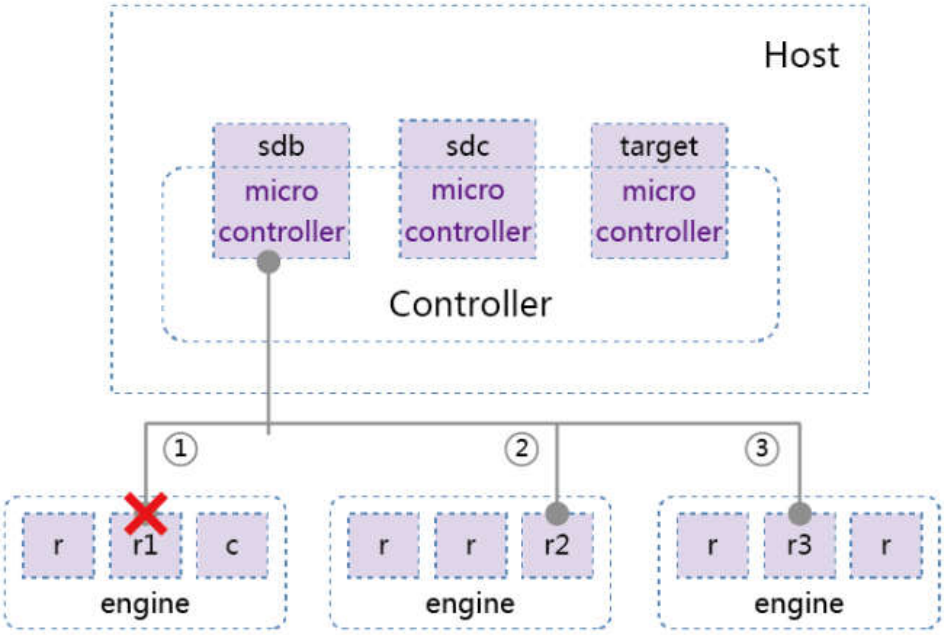
4.5.5 智能读 IO 处理

卷拥有多个副本的情况下，为了后端 engine 的负载均衡，需要将读 IO 请求，分散到多个副本中进行读取。如果是随机发送到多个 engine，往往是无法利用后端 engine 的预读功能，反而将读性能降低，VeSpace Server SAN 采取了简单的地址划分方式，将请求散到多个副本，每个副本的 cache 负责的地址空间缩小，导致缓存的命中率提升 n 倍(n 代表副本数量)，如下图：

如上图，replica 1 失效将由其他的 replica 接管其负责的地址，在上图中每个 replica 负责的地址空间范围相等，但是实际划分副本负责的地址空间范围是不一定相等的，副本的状态将影响其负责的地址空间范围，如副本与卷在同一台机器，副本所在的机器性能高等因子。

当某个副本失效时，如下图副本 r1 失效，此时发送给 r1 的 IO 请求将会返回错误，此时触发 Tunnel 层的错误处理，Tunnel 负责将副本负责的地址空间映射修复后，重试该请求，并将正确的请求结果返回给卷，这对使用卷的应用程序时完全无感知的。

版权所有 ©深圳市云舒网络技术有限公司



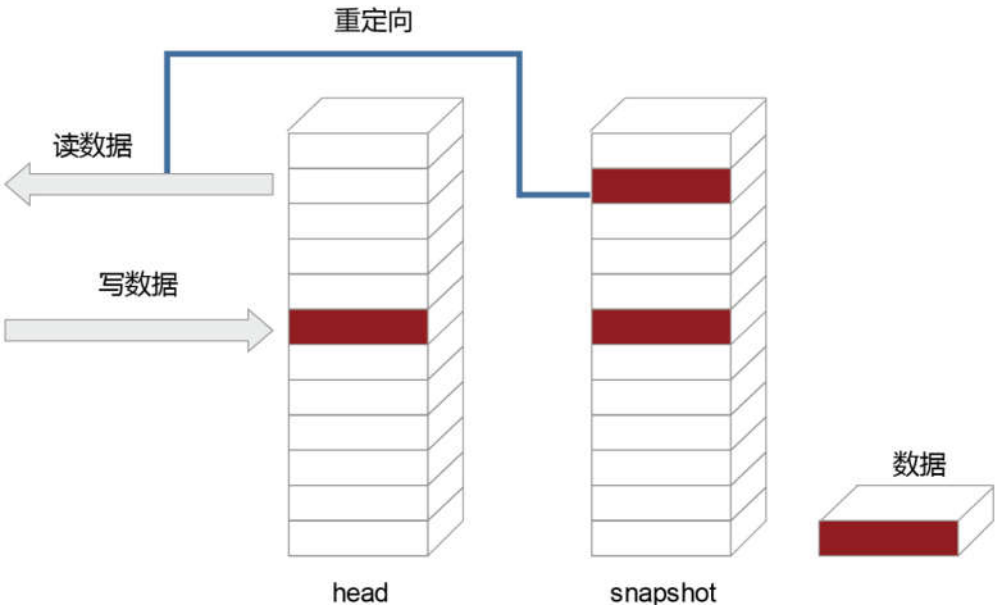
4.6 快照

VeSpace Server SAN 提供了快照机制，将用户的卷数据在某个时间点的状态保存下来，后续可以作为导出数据、恢复数据之用。当存储设备发生应用故障或者文件损坏时可以进行快速的数据恢复，将数据恢复到某个可用的时间点的状态。快照也可以为存储用户提供另外一个数据访问通道，当原数据进行在线应用处理时，用户可以访问快照数据，还可以利用快照进行测试等工作。VeSpace Server SAN 支持用户以只读的方式直接访问快照数据，例如直接只读挂载一个快照卷。

VeSpace Server SAN 快照数据在存储时采用 I/O 重定向（I/O Redirect），即将读写操作重新定向到另一个存储空间中。在一个快照生成期间，所有的写操作将被重定向到另一个介质，而读操作是否需要读重定向，则需要根据读取的位置是否有过自上次快照以来的写重定向，必须对有过写重定向的位置进行读重定向，否则不需要进行读重定向。与写时复制（CopyOn Write,COW）相比，写操作减少了一次数据复制的过程，大大提高了 IO 性能，并且性能不受快照数量影响。读操作因为需要重定向，会有一点性能开销，但如果存储节点有多余的内存，VeSpace Server SAN 会自动建立快照索引，使读性能大大提高。

快照在创建和回滚时只需修改元数据信息，而不用修改底层的数据，所以可以做到毫秒级创建和回滚。

版权所有 ©深圳市云舒网络技术有限公司

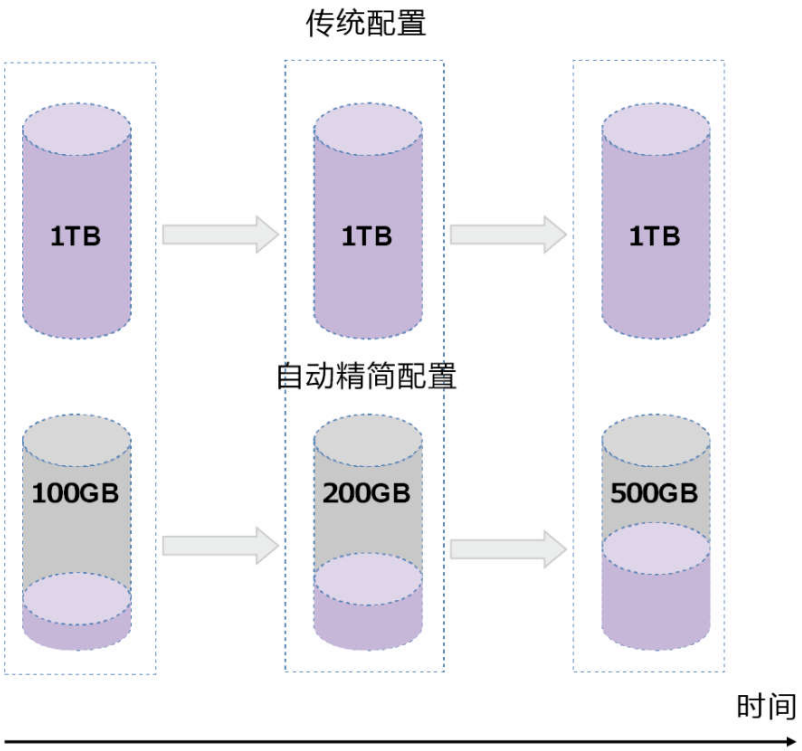


4.7 自动精简配置

VeSpace Server SAN 提供了自动精简配置功能，为应用提供比实际物理存储更多的虚拟存储资源。采用自动精简配置技术的数据卷分配给用户的是一个逻辑的虚拟容量，而不是一个固定的物理空间，只有当用户向该逻辑资源真正写数据时，才按照预先设定好的策略从物理空间分配实际容量。自动精简配置实现了将逻辑空间和物理空间分离的虚拟化容量分配技术，它不仅解决了单个应用的初始空间分配和扩容的难题，还大大提高了整个存储系统的资源利用率。利用已有的文件系统特性，VeSpace Server SAN 无需使用专门的元数据来记录卷的精简分配情况，和传统 SAN 相比，不会带来性能下降。

如果采用传统的磁盘分配方法，需要用户对当前和未来业务发展规模进行正确的预判，提前做好空间资源的规划。但这并不是一件容易的事情，在实际中，由于对应用系统规模的估计不准确，往往会造成容量分配的浪费，比如为一个应用系统预分配了 1TB 的空间，但该应用却只需要 100GB 的容量，这就造成了 900GB 的容量浪费，而且这 900GB 容量被分配了之后，很难再被别的应用系统使用。即使是最优秀的系统管理员，也不可能恰如其分的为应用分配好存储资源，而没有一点的浪费。

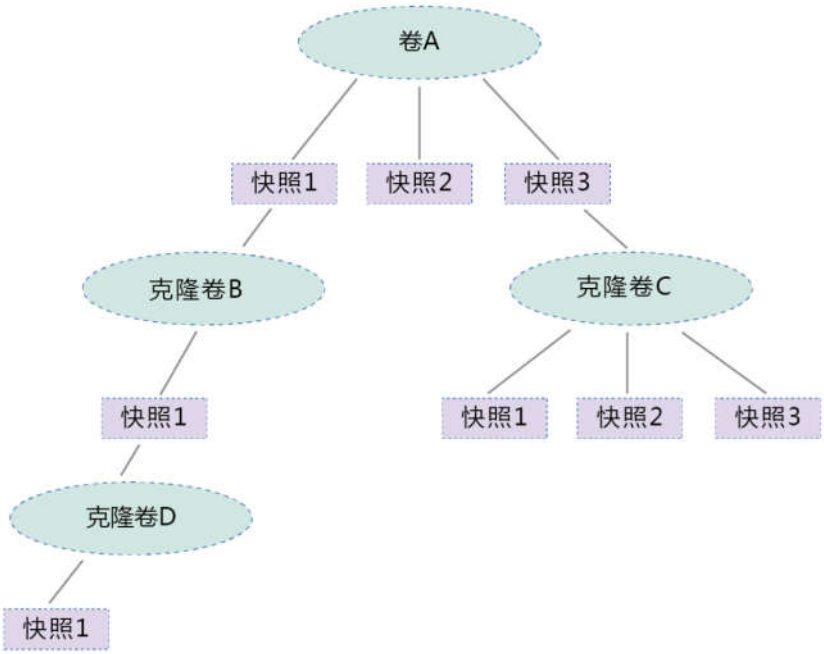
版权所有 ©深圳市云舒网络技术有限公司



4.8 卷克隆

VeSpace Server SAN 提供克隆机制，克隆支持链接克隆和拷贝克隆。克隆是基于卷快照，一个卷快照可以创建出多个克隆卷，克隆卷也可以继续克隆出下一层次的克隆卷。克隆卷刚创建出来时的数据内容与卷快照中的数据内容一致，后续对于克隆卷的修改不会影响到原始卷的快照和其他克隆卷。克隆时支持链接克隆，克隆卷和原始卷使用部分相同的底层数据，大大加快了克隆的速度和提升存储空间利用率。但如果克隆次数过多，由于克隆卷和原始卷使用的是相同的底层数据，这样就会造成数据分布不均衡和 IO 访问局部过热。所以当克隆数过大时， 策略子系统会使用拷贝克隆，把数据拷贝到其他节点，从而实现数据均衡。

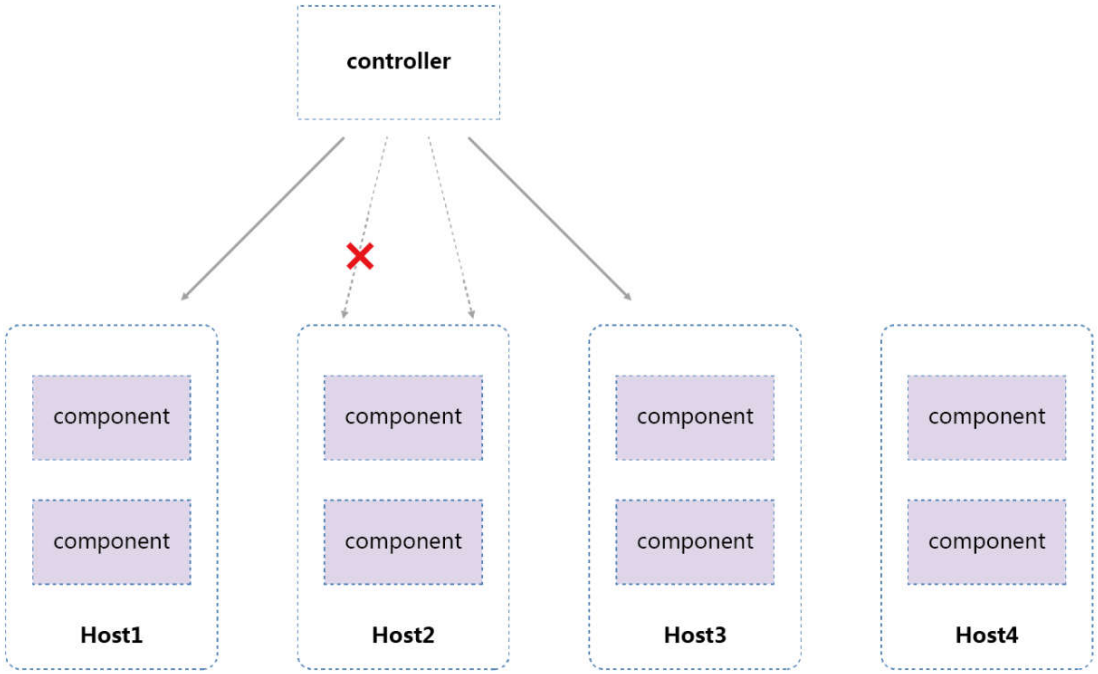
版权所有 ©深圳市云舒网络技术有限公司



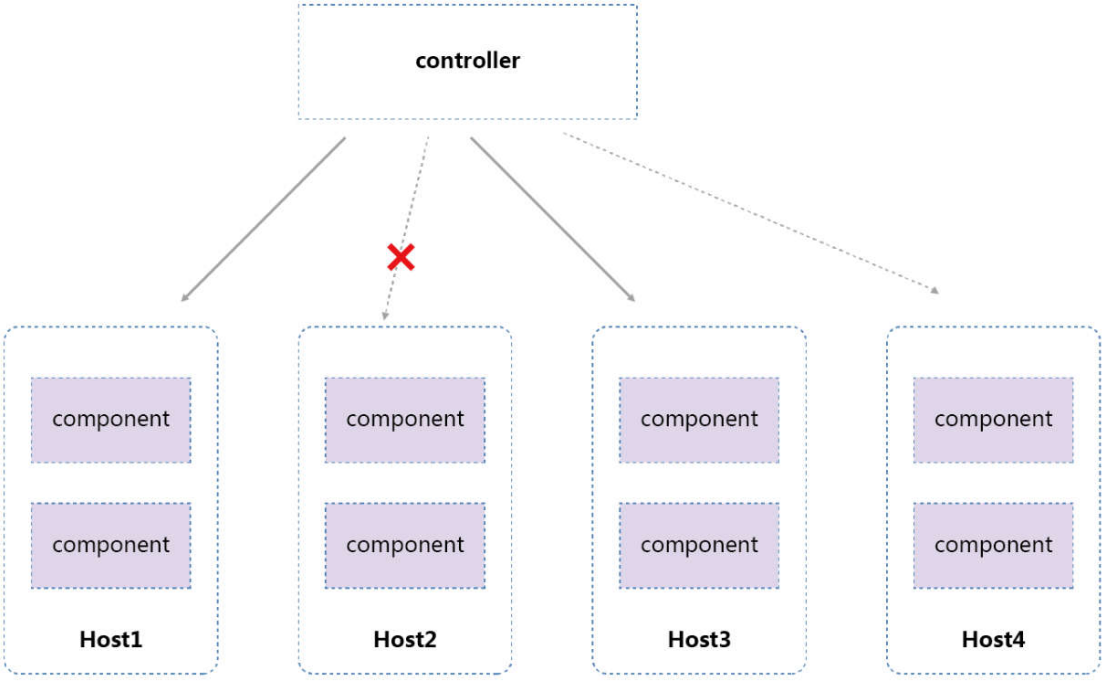
4.9 数据重建

当节点或者磁盘发生故障时，会自动进行数据重建。例如当节点发生故障时，存有故障节点数据备份的健康节点会以 component 为单位进行数据重建。该过程是多对多的拷贝方式， 这种机制可以充分保证数据重建的速度。重建分为局部重建和完全重建两种情况。在某些特定情况下，节点或者磁盘可能会发生短暂离线，例如网络闪断。副本只是丢失几个 IO，此时应该进行快速的局部重建，局部重建会计算数据哈希值，减少不必要的数据传输。如果当节点或者磁盘发生故障时间超过设置的阈值，此时需要进行完全重建，把副本数据拷贝到其他健康节点，此时会跳过计算数据哈希值，直接进行数据重建。VeSpace Server SAN 数据重建利用了毫秒级创建快照的特点，在数据重建之前，先对 component 创建一个快照，此时副本之间只是快照数据不同，而快照数据是只读的，所以对数据进行重建时可以不用考虑前端应用的读写 IO，大大简化了重建的过程，提高了重建的速度和保证了数据的一致性。

版权所有 ©深圳市云舒网络技术有限公司



局部重建

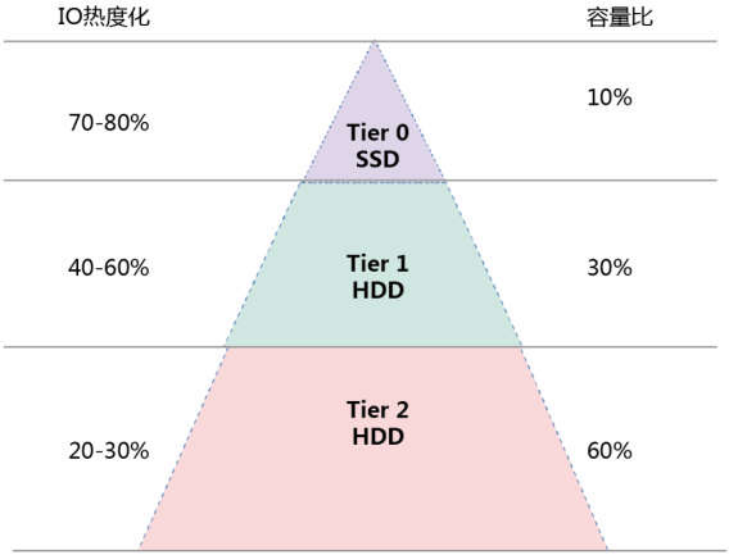


完全重建

版权所有 ©深圳市云舒网络技术有限公司

4.10 分层存储

自动存储分层（Automated Storage Tier, AST）管理系统的基本业务是能够将访问热度较低的数据安全地迁移到较低性能的存储层，将访问热度较高的数据迁移到更高性能的存储层中。如果用户在持久化存储存储层中混合使用了 SSD 和 HDD，VeSpace Server SAN 会自动把 SSD 和 HDD 分成两层，并启用 IO 热度统计，每隔一段时间，根据 IO 热度，把 IO 热度高的 component 迁移到 SSD 层，把 IO 热度低的 component 迁移到 HDD 层。用户也可以根据 HDD 的性能，把 HDD 分为高性能 HDD 和低性能 HDD，或者把 SSD 分为 PCI-E SSD 和 SATA/SAS SSD，这样可以自定存储分层的层数。策略子系统为用户提供比较灵活的配置选项来设置自动存储分层，用户可以根据前端应用场景，设置卷所在的层次，例如对于一些性能要求比较高的卷，可以把卷设置为 SSD 层。



4.11 多级 Cache 机制

VeSpace Server SAN 后端存储采用多级 cache 机制提升存储 IO 性能，读、写 cache 机制采用不同流程，读 cache 采用内存和 SSD 两级 cache 机制，写 cache 要保证数据不丢失，所以只使用 SSD 一级 cache。默认情况下，SSD 缓存的 70%为读缓存，用于存储频繁读取的磁盘块，从而最大限度减少对速度缓慢的磁盘的访问。缓存的 30%为写缓存，用于执行写入操作，每个 IO 会先写入缓存层，再批量写入低速磁盘层。

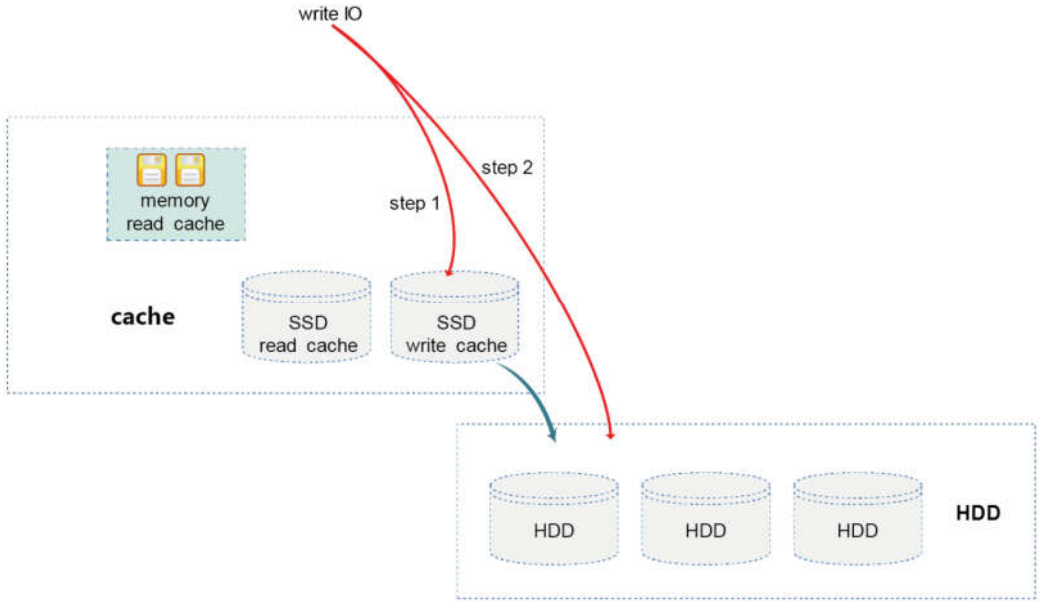
Write cache 机制：

版权所有 ©深圳市云舒网络技术有限公司

后端存储在收到应用端发送的写 IO 操作时，会将写 IO 缓存在 SSD cache 后完成本节点写操作。同时，周期性的将缓存在 SSD cache 中的写 IO 数据批量写入到硬盘，写 cache 有一个水位值，未到刷盘周期超过设定水位值也会将 cache 中数据写入到硬盘中。

VeSpace Server SAN 支持大块直通，按缺省配置大于 256KB 的块直接落盘不写 cache，这个配置可以修改。

写操作做了专门优化， 通过将随机的小 I/O 写请求合并成一个大 I/O 写请求， 然后顺序写到 SSD cache 中， 从而大大提升了 I/O 吞吐量。SSD cache 中的写 IO 数据刷到磁盘时，会合并顺序 IO，从而再次最大限度提高磁盘性能。



Read cache 机制：

VeSpace Server SAN 后端存储的读缓存采用分层机制，第一层为内存 cache，内存 cache 采用改进的 LRU 机制缓存数据，具有 LRU 的优点，同时能避免当发生全盘数据备份、全盘查找等此类扫描式操作时，缓存空间会被迅速挤占耗尽，原本存放在缓存中的热数据则被交换出缓存的缺点。第二层为 SSD cache，SSD cache 采用热点读机制，系统会统计每个读取的数据，并统计热点访问因子，当达到阈值时，系统会自动缓存数据到 SSD 中，同时会将长时间未被访问的数据移出 SSD。同时 VeSpace Server SAN 支持预读机制，统计读数据的相关性，读取某块数据时自动将相关性高的块读出并缓存到 SSD 中。

后端存储在收到应用端发送的读 IO 操作时，会进行如下步骤处理：

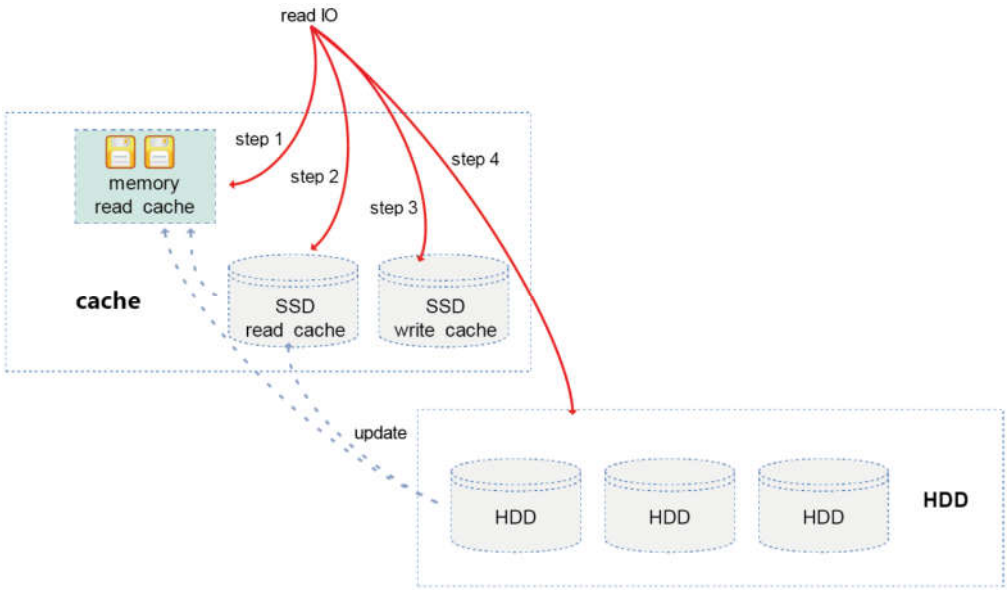
Step 1：从内存“读 cache”中查找是否存在所需 IO 数据，如果存在，则直接返回，同时更新 LRU 队列，否则执行 Step 2；

版权所有 ©深圳市云舒网络技术有限公司

Step 2：从 SSD 的“写 cache”中查找是否存在所需 IO 数据，如果存在，则直接返回，同时增加该 IO 数据的热点访问因子 如果热点访问因子达到阈值 则会被缓存在 SSD 的“读 cache”中。如果不存在，执行 Step 3；

Step 3：从 SSD 的“读 cache”中查找是否存在所需 IO 数据，如果存在，则直接返回，同时增加该 IO 数据的热点访问因子，否则执行 Step 4；

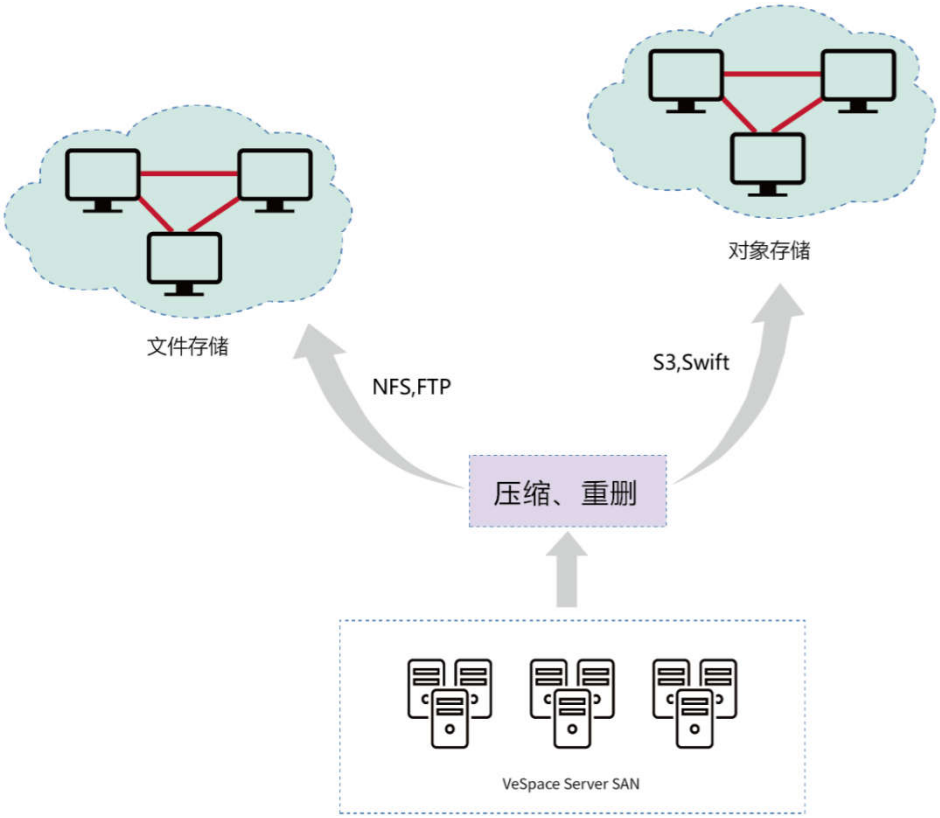
Step 4：从硬盘中查找到所需 IO 数据并返回，同时增加该 IO 数据的热点访问因子，如果热点访问因子达到阈值，则会被缓存在 SSD 的“读 cache”中。



4.12 数据卷备份

数据备份是容灾的基础，是指为防止系统出现操作失误或系统故障导致数据丢失，而将全部或部分数据集合从应用主机的硬盘或阵列复制到其它的存储介质的过程。VeSpace Server SAN 支持以卷为单位进行数据备份。备份是基于复制数据管理（Copy Data Management，CDM）技术，除了第一次备份时是全量备份之外，以后都是采用增量备份。备份属于块级备份，通过把卷拆分成多个小文件（默认为 1 M）进行备份。对于每个小文件，进行备份之前，先计算数据的哈希值，如果该哈希值已存在，说明备份系统中已存在相同的数据块，这时只需保存数据块的元数据，达到重复数据删除的作用。小文件进行备份时会对数据块进行数据压缩，更高效地使用存储资源，节省备份成本。数据备份支持对象存储和文件存储，对象存储使用 S3 和 Swift 接口，文件存储使用 NFS，FTP 接口。

版权所有 ©深圳市云舒网络技术有限公司



4.13 数据迁移

VeSpace Server SAN 监控系统会不间断的监控存储资源池的利用率，并在多台存储节点之间智能地分配存储资源。在某种极端情况下，可能会造成某个存储节点的使用率过高。此时策略层将会触发数据迁移，将某个或某几个过载节点上的副本数据迁移到集群中其他相对空闲的节点上，实现存储资源的动态分配，从而实现负载均衡。数据迁移跟数据重建一样，也是以 component 为单位，并且优先选择没有被使用的 component 进行迁移，保证了数据迁移基本不影响前端应用的读写 IO。

4.14 数据加密存储

未经任何加密的用户数据上传保存在云存储系统中,一旦数据泄露必然造成用户隐私的泄露。为了有效地保护企业用户和私人用户数据的安全，VeSpace Server SAN 支持全方位的数据加密保护。用户可以对某些保存了关键信息的数据卷开启数据加密保护功能。加密包括传输加密和全盘数据加密，既保证了敏感数据不会被外部黑客窃取又保证了硬盘遗失或者维护过程中数据不被泄露。

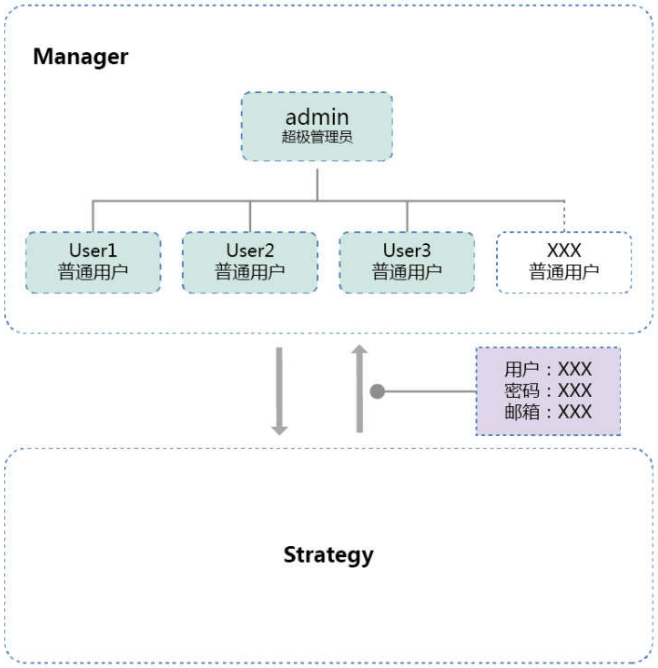
版权所有 ©深圳市云舒网络技术有限公司

五、VeSpace 管理子系统

5.1 用户

VeSpace Server SAN 设立超级管理员和普通用户的管理机制，普通用户受超级管理员的管理。超级管理员拥有 VeSpace Server SAN 的所有管理权限，被授权的集群管理的普通用户可以对其授权的集群进行硬件管理和存储管理等操作，超级管理员对用户的管理分为四种：

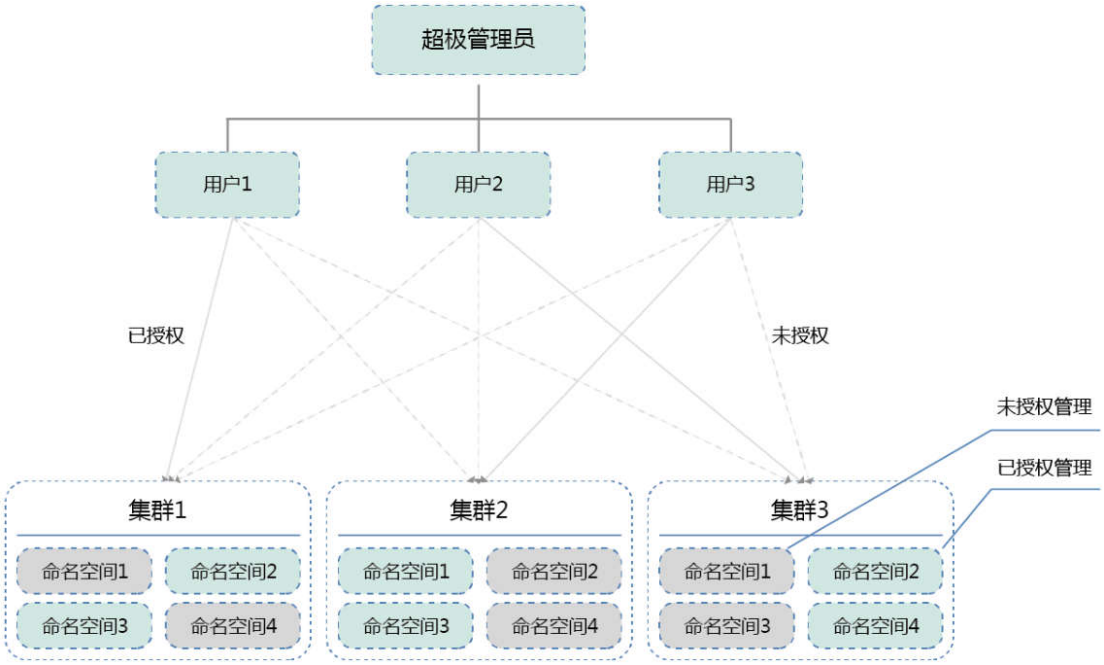
1.**添加用户**：超级管理员将用户资料（用户名、密码、邮箱）通过 manager 向后端发送添加用户请求，请求成功后用户添加成功，添加的用户受超级管理员管理。



2.**删除用户**：超级管理员将用户名通过 manager 向后端发送删除用户请求，请求成功后该用户被删除，同时该用户失去对所有集群的管理和使用。

版权所有 ©深圳市云舒网络技术有限公司

3.**授权用户管理集群**：超级管理员可以授权普通用户管理一个或多个集群，授权管理集群成功后，再授权该集群下的命名空间，普通用户才可以正常使用并管理该集群。



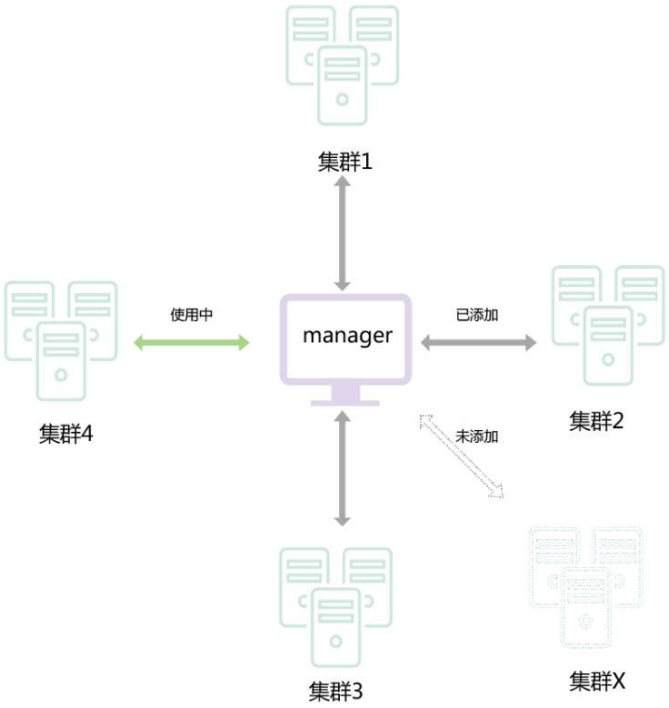
4.**取消用户管理集群**：对已授权管理集群的用户，超级管理员可以取消其对集群的管理权限，取消成功后，该用户不能再使用并管理该集群。

5.2 多集群管理

VeSpace Server SAN 允许超级管理员添加多个集群进行管理，集群之间相互独立，互不影响。集群的操作有添加、删除、激活和使用，其中，普通用户不可以添加、删除或者激活集群。

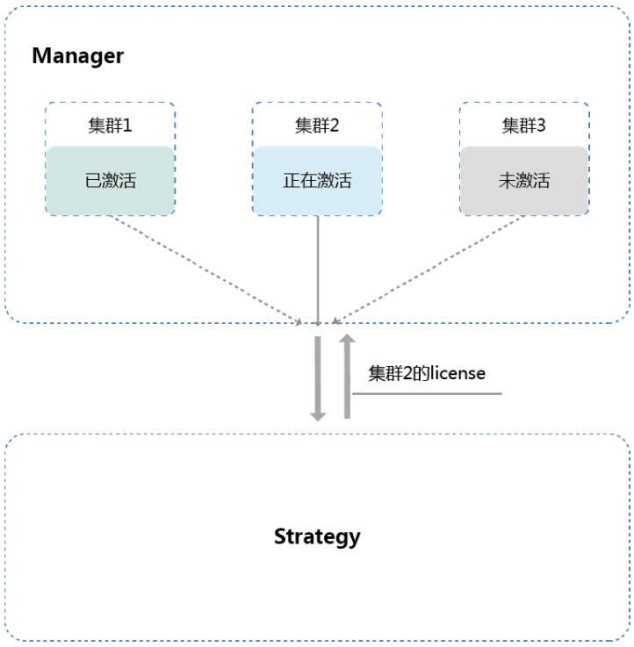
1.**添加集群**：超级管理员通过 manager 将集群名称、集群 IP 向后端发送添加集群的请求，请求成功后，集群添加至集群列表，超级管理员便可对其进行管理操作。

版权所有 ©深圳市云舒网络技术有限公司



2.**删除集群**：对不再管理的集群，超级管理员可以通过 manager 向后端发送删除集群的请求，请求成功后，该集群从集群列表中移除。

3.**激活集群**：对未激活的集群，不能进行硬件管理、存储管理中所有的添加和删除操作。超级管理员将 license 通过 manager 发送至后端进行验证，验证通过后激活成功，集群就可以正常使用和管理了。

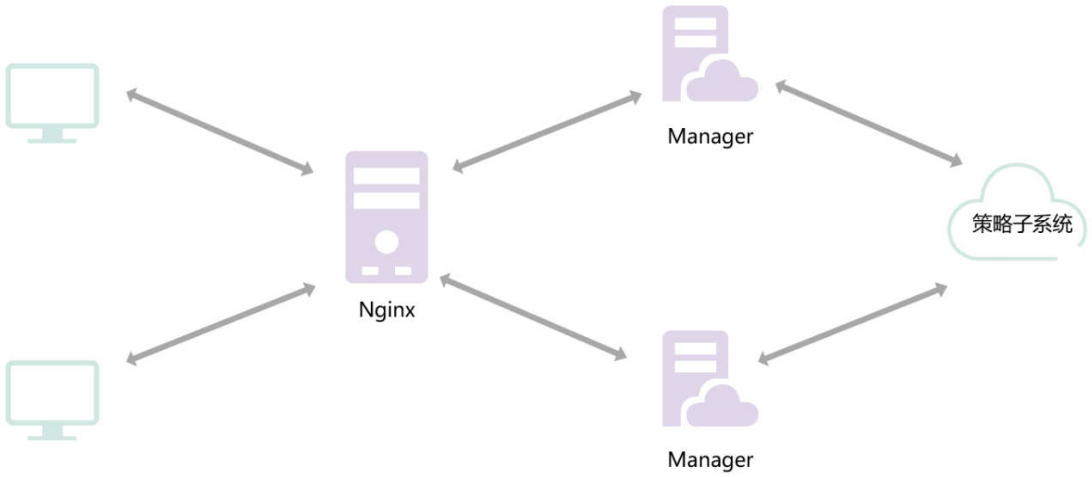


4.**使用集群**：用户进行通过切换集群，就可以选择集群进行硬件管理和存储管理等操作。

版权所有 ©深圳市云舒网络技术有限公司

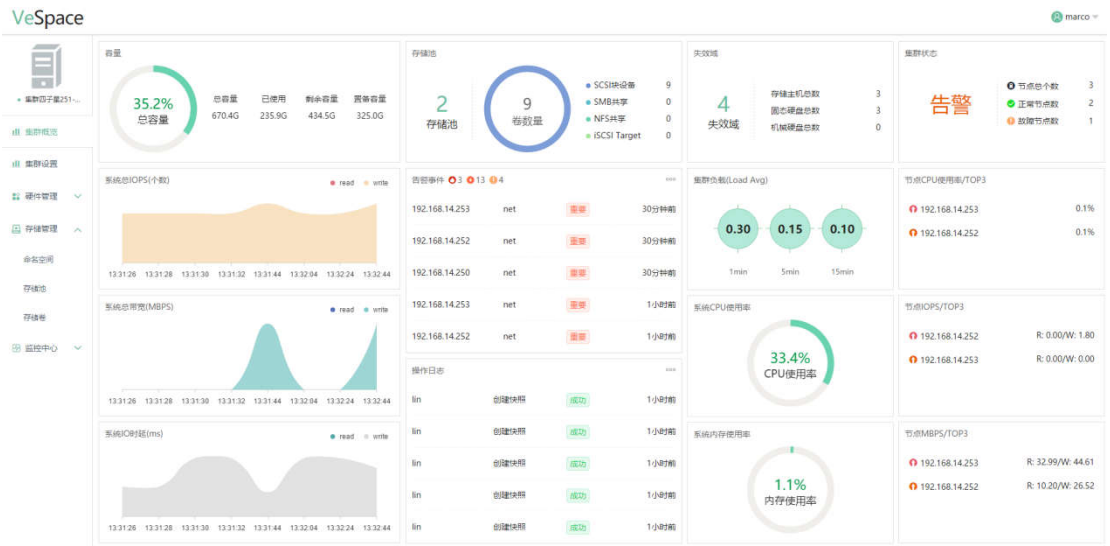
5.3 高可用负载均衡

后端部署两个 Manager 服务，用户通过 Nginx 反向代理访问 Manager 服务。当其中一台服务器宕机，Nginx 自动剔除不可用服务，从而实现不间断服务。



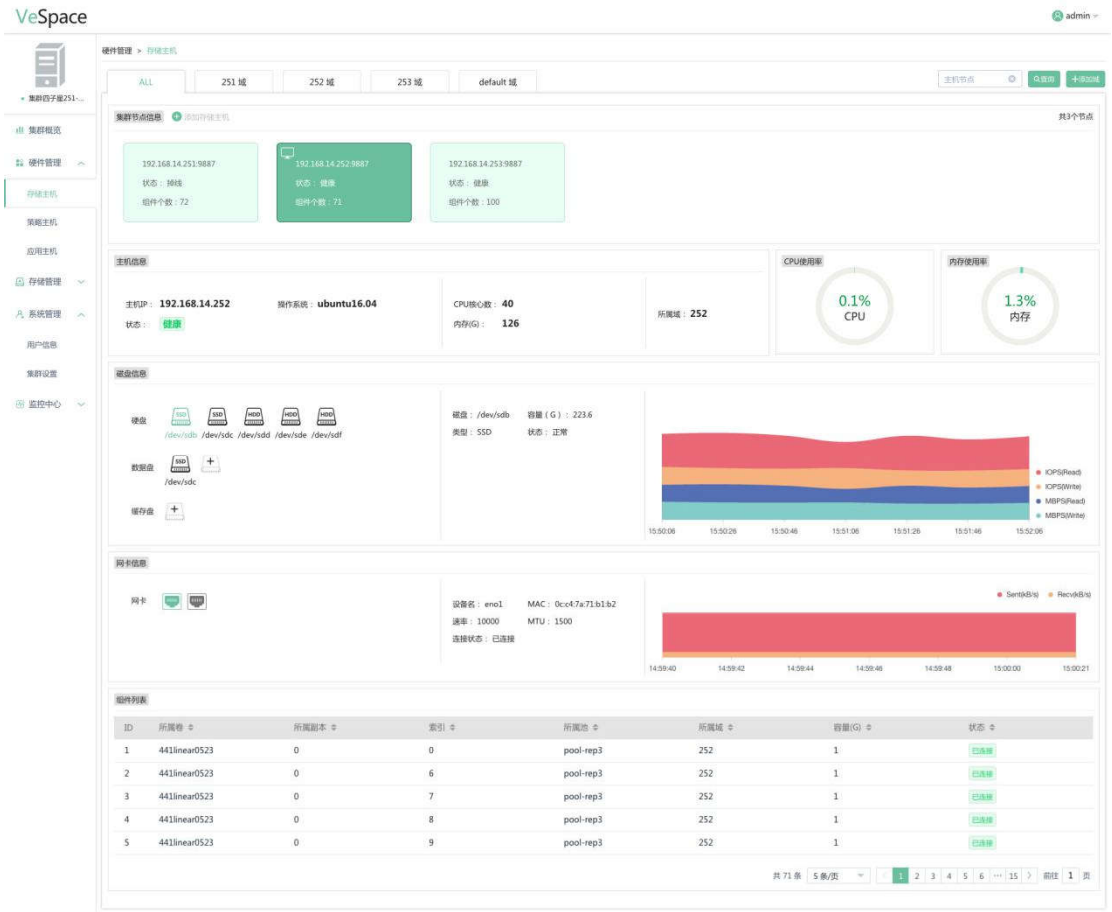
5.4 管理系统 UI 界面

5.4.1 集群概览

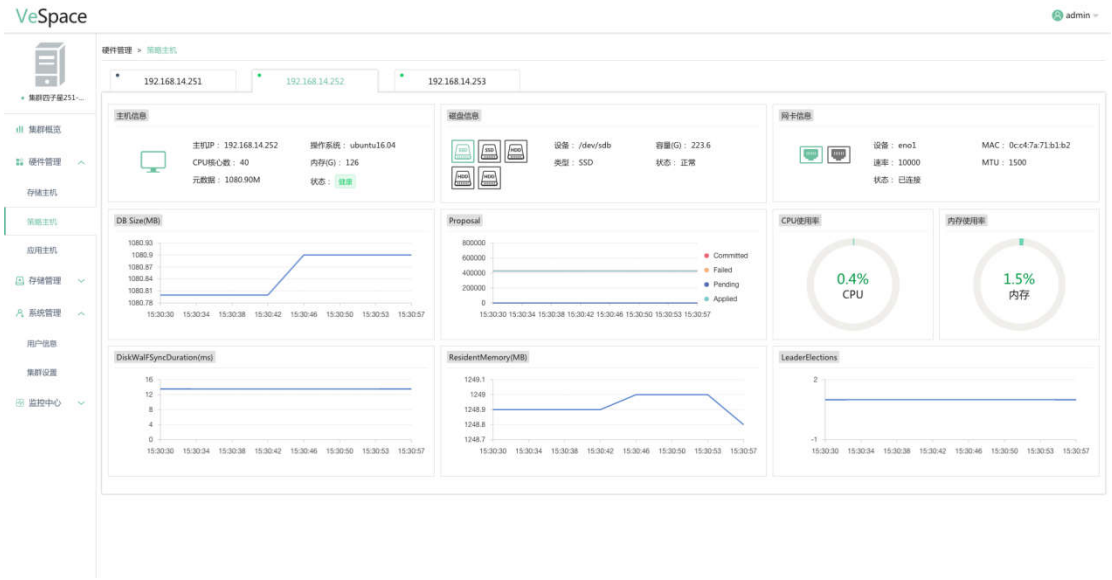


版权所有 ©深圳市云舒网络技术有限公司

5.4.2 硬件管理--存储主机

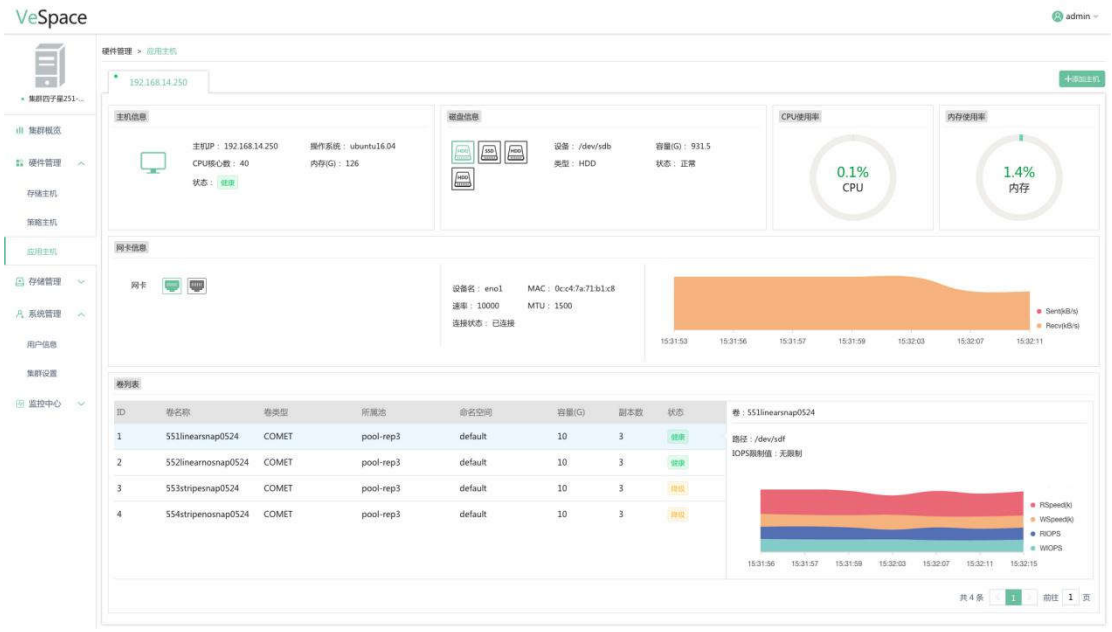


5.4.3 硬件管理--策略主机

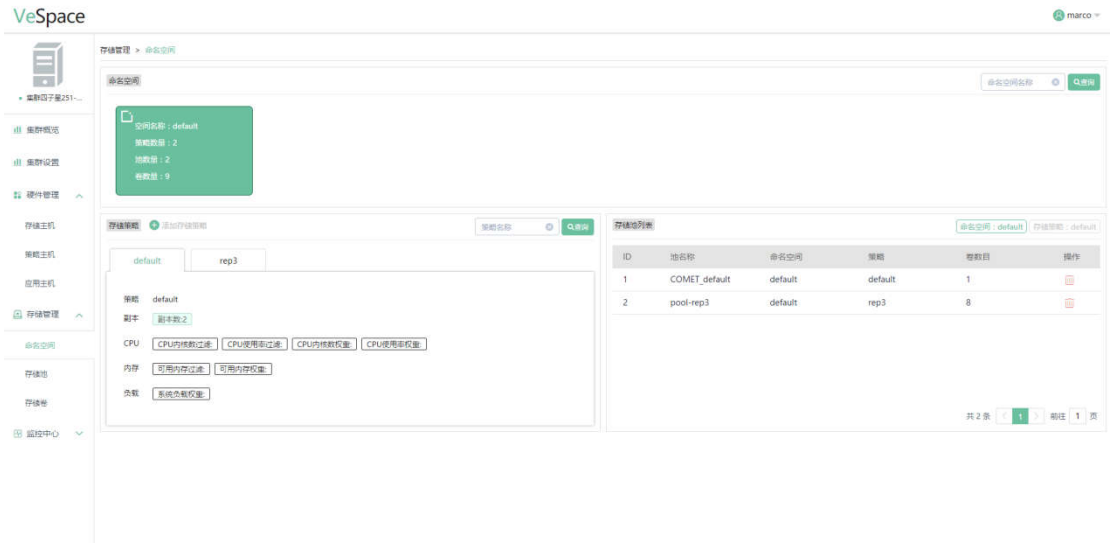


版权所有 ©深圳市云舒网络技术有限公司

5.4.4 硬件管理--应用主机

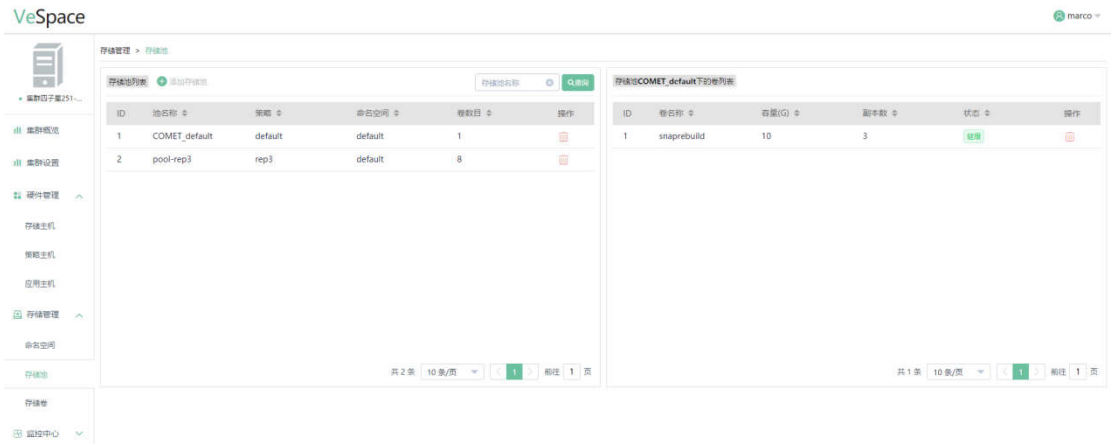


5.4.5 存储管理--命名空间

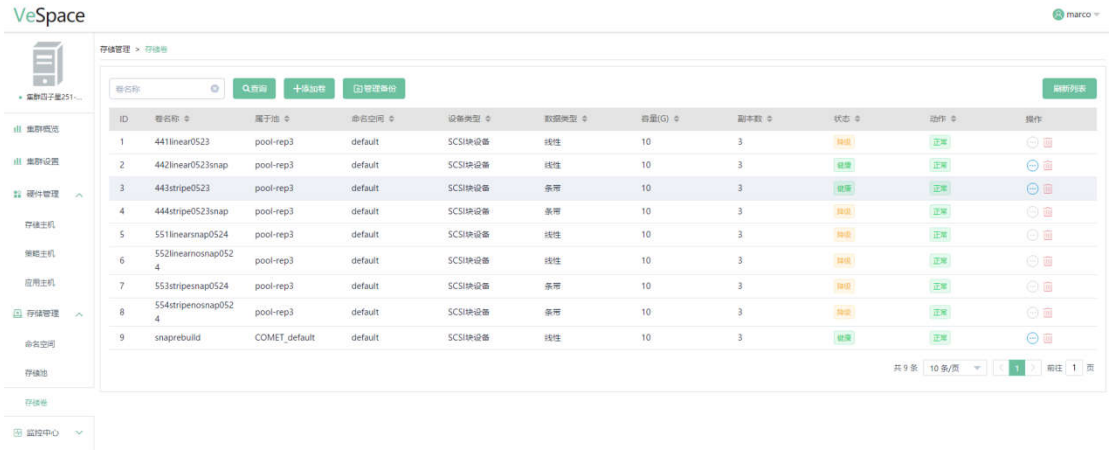


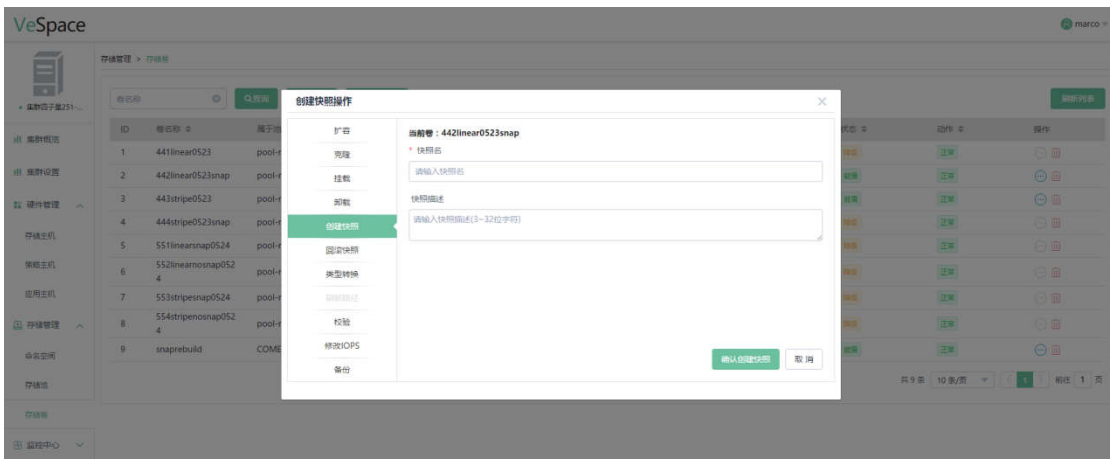
版权所有 ©深圳市云舒网络技术有限公司

5.4.6 存储管理--存储池

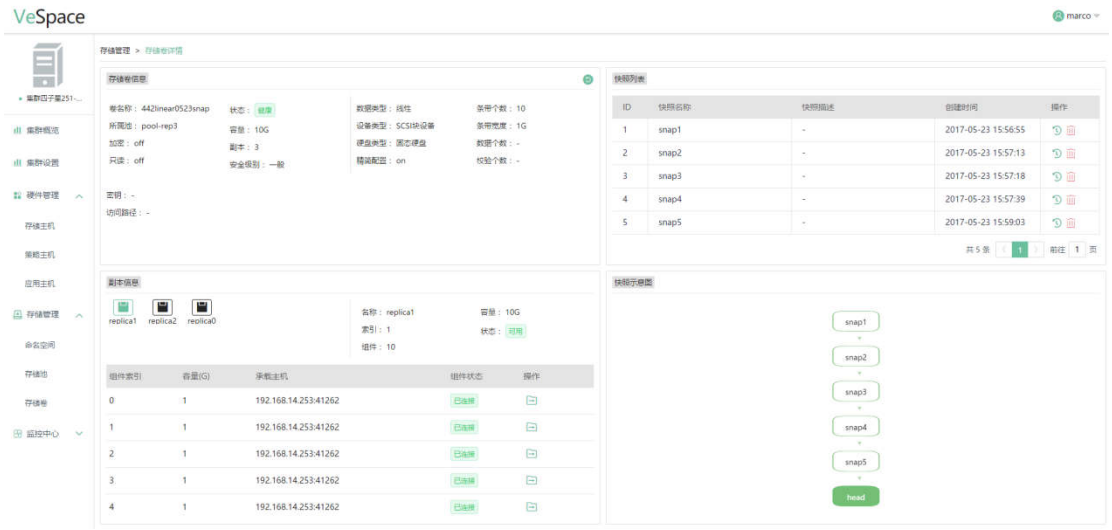


5.4.7 存储管理--存储卷

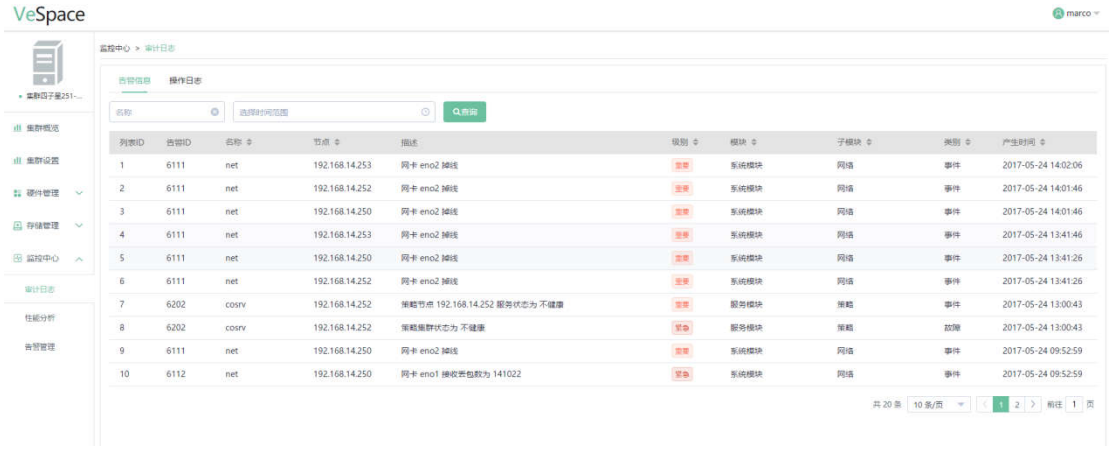




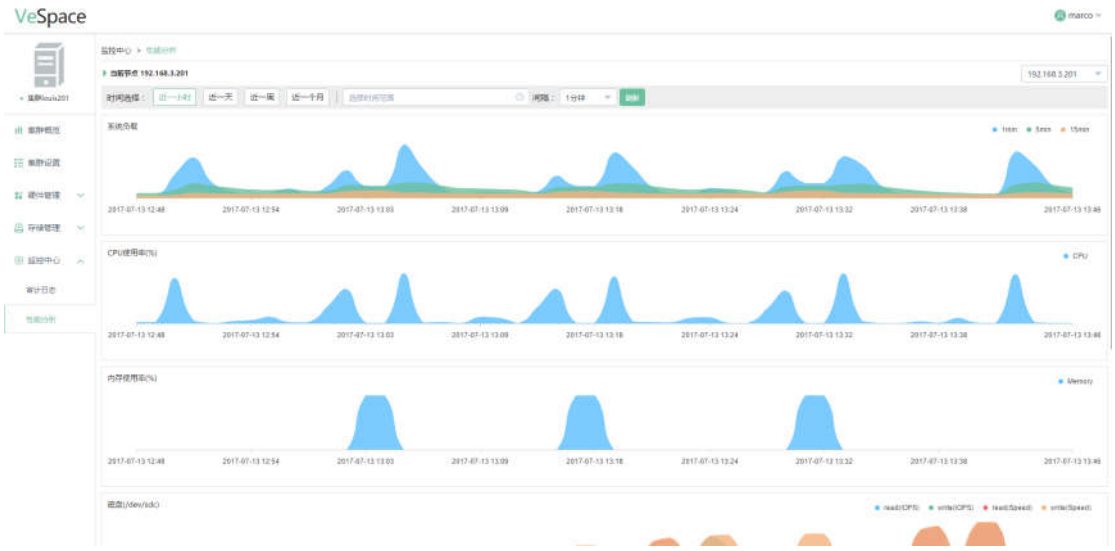
版权所有 ©深圳市云舒网络技术有限公司



5.4.8 监控中心--审计日志

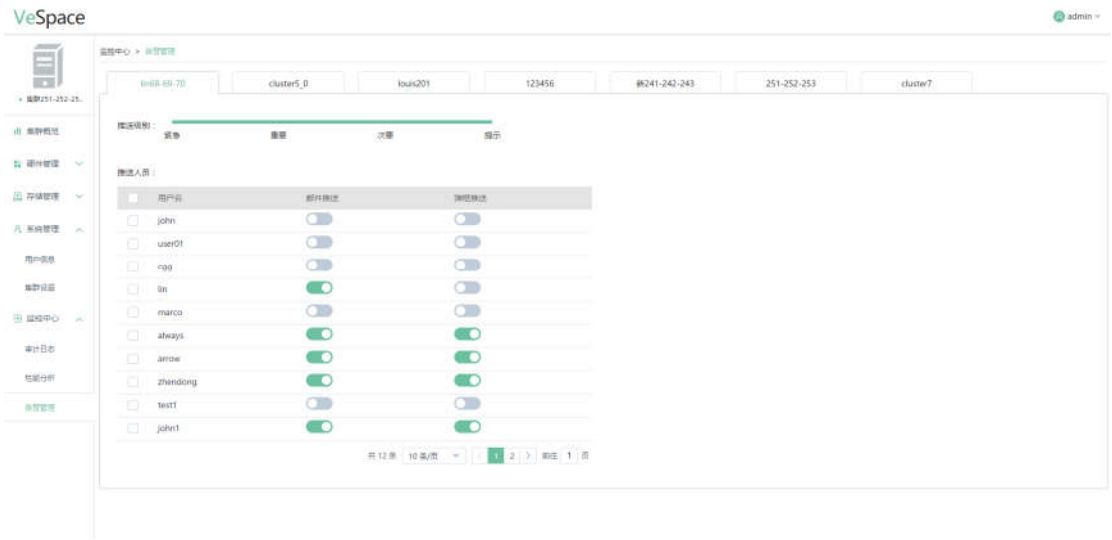


5.4.9 监控中心--性能分析



版权所有 ©深圳市云舒网络技术有限公司

5.4.10 监控中心--性能分析



六、兼容性

6.1 硬件平台

VeSpace Server SAN 产品对服务器硬件有如下基本要求：

- 通用 x86 服务器平台。
- 推荐 3 个及以上服务器，允许 ALL-IN-ONE 部署。
- 建议同一资源池中提供存储服务的服务器中的数据盘类型相同 ,规格和数量数量最好相同。
- 每台存储服务器若使用 SAS 盘或 SATA 盘做主存 ,可选配置 PCI-E SSD 卡或 SSD 盘用作 cache。
- 建议每个存储主机预留内存(4GB*磁盘个数) ,每个应用主机预留内存(2GB*Volume 个数) 用于运行 VeSpace Server SAN。
- 建议每服务器提供 4Gb 带宽用于 VeSpace Server SAN 通信，推荐使用万兆网络。用于融合部署云资源池 ,且超过 16 个节点时 ,业务平面和存储平面必须使用独立的网卡。

6.2 操作系统

VeSpace Server SAN 产品支持目前主流 Linux OS： Ubuntu16.04/14.04， CentOS 6/7

版权所有 ©深圳市云舒网络技术有限公司

6.3 规格参数

指标名称	指标值
单集群最大保护域数	64
单集群最大命名空间数量	4095
单集群最大存储池数量	1048576
单集群最大存储主机数量	4096
单集群硬盘数量	65536
单集群支持的策略主机数量	3， 5， 7
单集群支持的应用主机数量	102400
单集群最大卷数量	1048576
单命名空间容纳存储池最大数量	4096
单存储池容纳存储卷的最大数量	10240
存储卷最大容量	24TB
每 TB 数据恢复时间	HDD: 60 分钟 SSD: 30 分钟
数据冗余模式	n 副本(1<n<16), n+m 纠删码
存储卷快照数量	255
存储卷容量范围	128MB~24TB

6.4 缩略语表

英文缩写	英文全称	中文全称
SDS	Software Defined Storage	软件定义存储
LUN	Logical Unit Number	逻辑单元号

版权所有 ©深圳市云舒网络技术有限公司

RAID	Redundant Array of Independent Disks	独立磁盘冗余阵列
SAS	Serial Attached SCSI	串行 SCSI
SATA	Serial Advanced Technology Attachment	串行 ATA
SSD	solid state disk	固态硬盘
SCSI	Small Computer System Interface	小型计算机系统接口
OLTP	On-Line Transaction Processing	联机事务处理系统
OLAP	On-Line Analytical Processing	联机分析处理系统
SAN	Storage Area Network	存储域网络
NAS	Network Attached Storage	网络附属存储
Server SAN	Server SAN	服务器 SAN 存储

版权所有 ©深圳市云舒网络技术有限公司